IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Appln. Of:	KOTAKA
Serial No.:	10/090,302
Filed:	March 4, 2002
For:	ARITHMETIC OPERATION METHOD FOR CYCLIC . . .
Group:	2133
Examiner:	Dipakkumar Gandhi DOCKET: NEC N01321

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 CFR 1.131 OF PRIOR INVENTION IN A WTO MEMBER COUNTRY TO OVERCOME A CITED PATENT REFERENCE

Dear Sir:

The undersigned, being the named inventor of the subject application, declares and states the following:

(1) I conceived of and completed the invention described and claimed in the subject application, in Japan, prior to January 26, 2001, the U.S. filing date of the Ishiwaki U.S.Patent No. 6,725,415 cited in the Office Action mailed September 24, 2004 in the above matter.

(2) As proof thereof, I provide the following:

(a) Exhibit A, which is a full and complete copy of a written Invention Disclosure, which I prepared and submitted to the IP Division of NEC Electronics Corporation, the assignee of the subject application. As can be seen, the drawing figures attached to Exhibit A essentially correspond to the drawing figures submitted with the subject application. A verified English translation of Exhibit A also is attached hereto.

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

(b) My Invention Disclosure was accepted for filing by NEC Electronics Corporation, and a detailed description was then supplied to an outside Japanese Patent Law Firm, who then prepared the documents for filing a Japanese patent application. The application was prepared, reviewed by me, and filed in the Japanese Patent Office as Japanese Patent Application Serial No. 2001-059807 filed March 5, 2001.

(3) The foregoing and attachments clearly show a date of conception and completion of the invention of this application all prior to the January 26, 2001 U.S. filing date of the Ishiwaki U.S. patent application. Moreover, having conceived of the invention prior to January 26, 2001, I proceeded diligently to prepare a complete written disclosure of same and to then promptly file a patent application, initially in Japan, and thereafter, in the United States, covering the invention. At no time between my conception of the invention, and my filing of the subject U.S. Patent Application, did I ever intend to abandon the invention.

As the named inventor, I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Shigenari Kotaka

Shigenari KOTAKA

Date December 24, 2004

IYES SOLOWAY P.C.
1 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
NCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Appln. Of: KOTAKA

Serial No.: 10/090,302

Filed: March 4, 2002

For: ARITHMETIC OPERATION METHOD FOR CYCLIC . . .

Group: 2133

Examiner: Dipakkumar Gandhi DOCKET: NEC N01321

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

VERIFICATION OF TRANSLATION

Dear Sir:

The undersigned hereby certifies that I am conversant in both Japanese and English languages, that I have prepared the attached English translation of the Japanese text attached as **Exhibit A**, and that the English translation is a true, faithful and accurate translation of the attached **Exhibit A**.

I further declare that all statements made of my own knowledge are true and that all statements made on information and belief are with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 USC § 1001, and that such false statements may jeopardize the validity of the application or any patent issuing therefrom.

Date: January 24, 2005 Signature: Yukuo Nishimura

Typed Name *Yukuo NISHIMURA*

Address: Nakamura The First Bldg. 3F. 4-2-6, Kitaurawa, Urawa-ku, Saitama, SAITAMA, Japan

HAYES SOLOWAY P.C.
130 W. CUSHING STREET
TUCSON, AZ 85701
TEL. 520.882.7623
FAX. 520.882.7643

175 CANAL STREET
MANCHESTER, NH 03101
TEL. 603.668.1400
FAX. 603.668.8567

TO: Nishimura International patent Office
Mr. Yukoo NISHIMURA

Date December 11, 2000

配布先 西村国際特許事務所 西村　征生　殿 参考：（1員のみ） 10-22440 知財部　高野マネージャー殿	コンカレント出願 明細書作成依頼書 Request for Preparing Specification	EDC知財－特出－1330 平成 12年 12月 11日 日本電気株式会社 エレクトロンデバイス知的財産部 マネージャー 和田　正 担当：久野　淑己 電話：044-435-1421 FAX：044-435-1871

FROM NEC Corporation

下記の件について明細書作成を依頼します。

－－－　記　－－－

1. 整理番号　：　754-10092
2. 提案名称　：　ＣＲＣ演算回路　CRC arithmetic Operation Circuit
   Title
   概要：infini bandのＣＲＣ演算においてレイテンシを削減するような回路
3. 提案者名　：　小高　重成　shigenari KOTAKA (三シス・一設G)
   Inventor
4. 当部担当者：　久野　淑己
5. 添付書類　：　アイデア提案書
   打ち合わせ記録
   特開平
   特開平

15日(金) 1:00～ 会議室

6. 希望納期　：　草稿－　平成12年01月13日
   出願－　平成12年01月20日
7. 打合せ場所：　玉川　地区を予定。
8. 備考：

注記

打ち合わせの記録を残し、明細書作成時の一助となるように「コンカレント打ち合わせ記録」を使うようにしました。（原則としてEDC知財部担当者が打ち合わせ時に記入）

この用紙にはEDC知財部と発明者とのｌｓｔ打ち合わせの内容がまとめて有り、２ｎｄ打ち合わせの内容もまとめられる様になっていますのでご活用下さい。尚、貴事務所においては、ｌｓｔ打ち合わせにとらわれることなく、プロの目から見た発明のとらえ方、ストーリの立て方、クレーム案等について事前に検討しておいて下さい。

IDEA PROPOSAL

Company Reference Number : 754-10092

Proposed Date : November 20,2000

グループコード: 部内番号:

Authorization] Manager 森 Chief 堀口 主任:

Semiconductor Engineering Department Received Date 知財部 12.11.30 沖山

Inventor Entry

E-mail: kotaka@lsi.nec.co.jp
Section: Third System L, First Design TEL:822-26011 メール番号:22-26010
Name Shigenari KOTAKA Company Code :0690257

適用・応用分野:通信

適用製品名: SUN Wings 売上規模: 3K (百万円／年)

実験・試作状況:○実験・試作完了 ●実験・試作中 ○実験・試作予定あり ○実験・試作予定なし

Prior art search (Patent): done (JP Appln. No.H02-119319)
Prior art search (Document): done, none

特許検索式:(CRC生成回路+CRC演算回路)&(高速+高速処理+レイテンシ+レイテンシー+Latency)

関連提案・特許:無し

サンプル出荷／社外発表予定:○無 ●有(早い方の日: H13年 9 月 日、何処で)

出願希望種別:●コンカレント ○S級 ○通常出願(届出予定日:~~1999~~2001年12月17日)

Chief Entry

上司氏名: 堀口立二

Decision of reduction to practice
Planning of Foreign Application: US)

コメント:

[発明相談コメント欄] センター担当: 年 月 日

[評価委員会記入欄]

評価責任者氏名: 古賀隆俊

決定日: 平2000年 11月27日

評価結果 1. 出願希望(●コンカレント ○S級 ○通常届出) 2. 公開技報 3. 中止 4. 再検討

外国出願希望:○無 ●有(国名:●米 ○韓 ○中国 ○台湾 ○英 ○独 その他)

届出指定日:平2000年 12 月 17 日

コメント: 計算方法を明確に. 米国出願ではアルゴリズムの権利化も検討して下さい. 再教トライでの処理例を追加し権利範囲を広げて下さい. 含む

センターへの要望:

Chief Signature

I read on pages 1 to 6 of this proposal and have understood this invention.
Name: Tatsuji HORIGUCHI November 20,2000

Inventor signature
Name Shigenari KOTAKA November 20,2000

氏名: 20 年 月 日

【CRC】

A CRC (cyclic redundancy check) is one of methods for checking whether or not data have correctly been transmitted (read or written) in data transmission, and in writing or reading data into/from a disk, a tape, or a like.

A CRC arithmetic operation is performed by using an expression made up of a combination of shift and addition, called a CRC generative polynomial. A value generally used in the CRC arithmetic operation is made up 16 bits or 32 bits (The word "cyclic" in CRC is derived from a calculation method in which power of 2 is used as a modulus, and an over-flow of an operational result is neglected).

Since the CRC arithmetic operation is not to perform a simple addition operation, there is demerit in that processing using a software for the CRC calculation greatly increases a processor workload. On the other hand, since processing by a hardware is simple and easy, the CRC arithmetic operation method is generally used in a disk controller, a communication LSI, and a like.

Source: ASCII digital term dictionary

【CRC generative polynomial】

The above-described CRC generative polynomials are defined as follows:

$$\text{CRC32(32bit)} : G(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$$

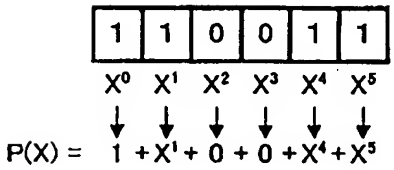
$$\text{CRC16(16bit)} : G(X) = X^{16}+X^{12}+X^{3}+X^{1}+1$$

【CRC method】

Next, the CRC arithmetic operation method will briefly be explained. The following, for simplicity's sake, example using 6 bits will be explained: because using 32 bits or 16 bits as an example makes the explanation (operational expressions) too complicated.

Please note that an arithmetic operation method using 32 bits or 16 bits is a same as that using 6 bits.

① A polynomial is given below, in which input data is considered to specify a value.



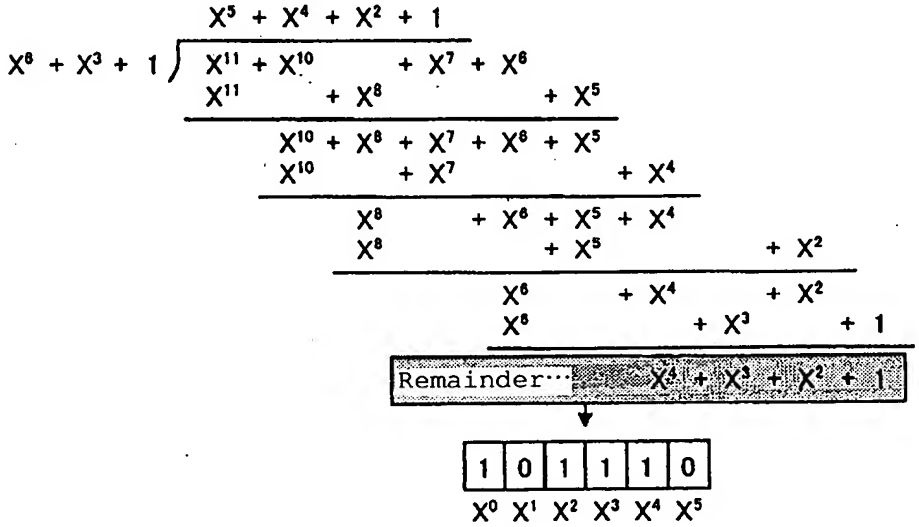
② Next, the CRC generative polynomial predetermined in a data transmitting/receiving is used.
(CRC32 and CRC16 generative polynomials are shown above.)

$$\text{CRC6(6bit)} : G(X) = X^6+X^3+1$$

③ A result obtained by multiplying the input data P(X) by the highest order term X^6 included in the generative polynomial G(X) is represented by Q(X).

$$Q(X) = X^{11}+X^{10}+X^7+X^6$$

④ Then, the Q(X) is divided by the generative polynomial G(X) and its remainder is used as a cyclic check bit of the CRC arithmetic operation method, which is called a "CRC code".



⑤ A new Q(X) is produced by multiplying input data to be input next by the CRC code obtained by the CRC arithmetic operation ④. By dividing the new Q(X) by the generative polynomial G(X), a new CRC code is obtained.

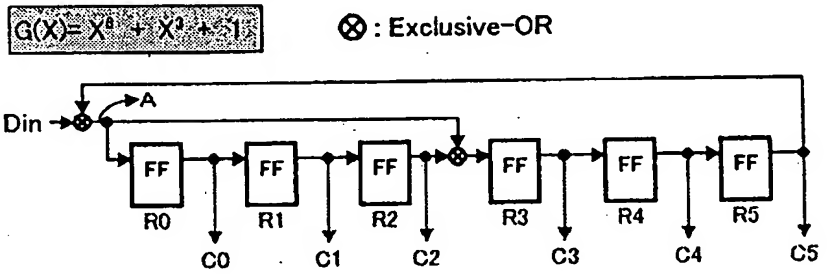
When the processing described above is performed repeatedly (in a cyclic manner) on all the input data, a CRC arithmetic operation result can be obtained, and the obtained CRC arithmetic operation result is transmitted by being added to an end of data to be transmitted.

【CRC operational expression】

The concept of the CRC arithmetic operation is as described above.

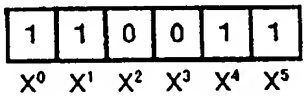
However, in a multi-bit CRC32 arithmetic operation such as the CRC32 arithmetic operation, the above-described division cannot be done simply by hardware because the hardware cannot perform high-speed processing or because large-sized circuits have to be used as the hardware and, therefore, the division is generally done using the following method. (Again, an example using 6 bits will be described.)

First, the following circuit can be obtained in accordance with the above-described CRC generative polynomial.



Next, when input data P(X) described earlier under the section "CRC method" is serially input through Din, an output state in each FF (flip-flop) will be given below:

-Input data-



Here, "A" denotes an Ex-OR of loop-back data and input data.

Shifted	Input value	A	FF output						Note
			C0	C1	C2	C3	C4	C5	
0	—	—	0	0	0	0	0	0	Initial value
1	1	1	1	0	0	1	0	0	
2	1	1	1	1	0	1	1	0	
3	0	0	0	1	1	0	1	1	
4	0	1	1	0	1	0	0	1	
5	1	0	0	1	0	1	0	0	
6	1	1	1	0	1	1	1	0	Remainder

Agreement can be seen between the remainder described in the above table and the remainder explained earlier under the section "CRC method", that is, a set of output data being output from each of the flip-flops when data are shifted sequentially by the number of its bits which is equal to that of the input data is the "CRC code" to be acquired.

The following operational expression can be introduced, wherein input data are expressed sequentially by D0-D5, each initial value of FFs is expressed by R0-R5, and each output of the FFs is expressed by C0-C5.

Shifted	Input Value	A	FF output	
0	—	—	C5	R5
			C4	R4
			C3	R3
			C2	R2
			C1	R1
			C0	R0
1	D5	R5·D5	C5	R4
			C4	R3
			C3	R2·R5·D5
			C2	R1
			C1	R0
			C0	R5·D5
2	D4	R4·D4	C5	R3
			C4	R2·R5·D5
			C3	R1·R4·D4
			C2	R0
			C1	R5·D5
			C0	R4·D4
3	D3	R3·D3	C5	R2·R5·D5
			C4	R1·R4·D4
			C3	R0·R3·D3
			C2	R5·D5
			C1	R4·D4
			C0	R3·D3
4	D2	R2·R5·D5·D2	C5	R1·R4·D4
			C4	R0·R3·D3
			C3	R5·D5·R2·R5·D5·D2
			C2	R4·D4
			C1	R3·D3
			C0	R2·R5·D5·D2

Note: "·"indicates EX-OR

Shifted	Input value	A	FF output	
5	D1	R1·R4·D4·D1	C5	R0·R3·D3
			C4	R5·D5·R2·R5·D5·D2
			C3	R4·D4·R1·R4·D4·D1
			C2	R3·D3
			C1	R2·R5·D5·D2
			C0	R1·R4·D4·D1
6	D0	R0·R3·D3·D0	C5	R5·D5·R2·R5·D5·D2
			C4	R4·D4·R1·R4·D4·D1
			C3	R3·D3·R0·R3·D3·D0
			C2	R2·R5·D5·D2
			C1	R1·R4·D4·D1
			C0	R0·R3·D3·D0

Note: "."indicates EX-OR

Operational expressions obtained by each of output data from the flip-flops, when data shift operations are performed six times (that is, six shifts), are CRC6 operational expressions. Here, since same terms (R3 · R3 etc.) can be deleted, the following operational expressions are what to acquired, by rearranging each of the obtained operational expressions.

CRC6 operational expressions

C5=R2·D2
C4=R1·D1
C3=R0·D0
C2=R2·R5·D2·D5
C1=R1·R4·D1·D4
C0=R0·R3·D0·D3

The following expressions are obtained, by inputting initial values "0" (R0-R5="0") and data (11011) to the above-mentioned operational expressions.

C5=0
C4=1
C3=1
C2=1
C1=0
C0=1

Thus, agreement can be seen between this result and the earlier result. Accordingly, it is confirmed that the operational expressions described above are effective to use for checking error.

The above is all of the method explanation and operational expression introduction in CRC6.

The operational expressions of CRC16 and CRC32 can also be introduced by using a same manner as described above in CRC6.

The generative polynomials, generating circuits and operational expressions of CRC16 and CRC32 will be shown below.

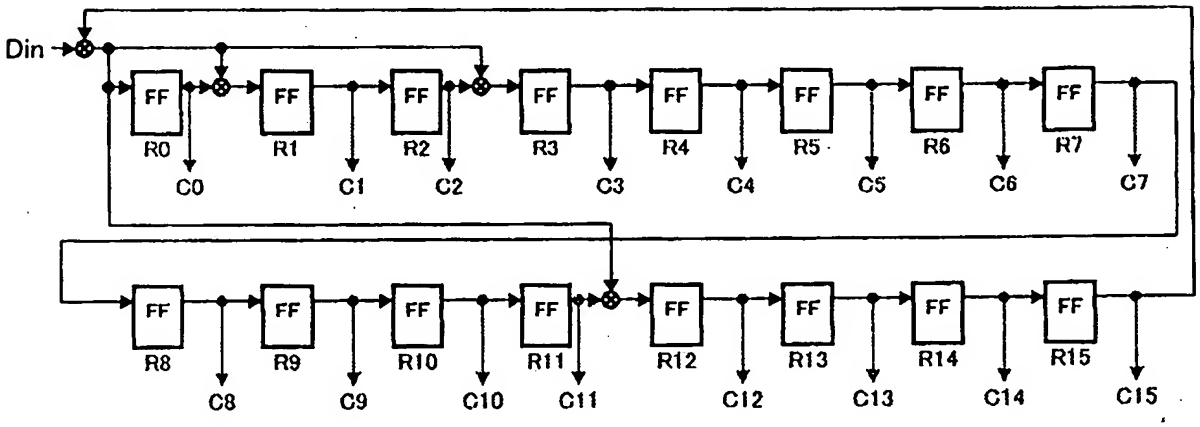
【CRC16】

Generative polynomial

$G(X) = X^{16}+X^{12}+X^{3}+X^{1}+1$

Generating circuit

⊗ : Exclusive-OR



Operational expressions

C0	R0·R4·R8·R12·R13·R15·D0·D2·D3·D7·D11·D15
C1	R0·R1·R4·R5·R8·R9·R12·R14·R15·D0·D1·D3·D6·D7·D10·D11·D14·D15
C2	R1·R2·R5·R6·R9·R10·R13·R15·D0·D2·D5·D6·D9·D10·D13·D14
C3	R0·R2·R3·R4·R6·R7·R8·R10·R11·R12·R13·R14·R15·D0·D1·D2·D3·D4·D5·D7·D8·D9·D11·D12·D13·D15
C4	R1·R3·R4·R5·R7·R8·R9·R11·R12·R13·R14·R15·D0·D1·D2·D3·D4·D6·D7·D8·D10·D11·D12·D14
C5	R2·R4·R5·R6·R8·R9·R10·R12·R13·R14·R15·D0·D1·D2·D3·D5·D6·D7·D9·D10·D11· D13
C6	R3·R5·R6·R7·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D8·D9·D10·D12
C7	R4·R6·R7·R8·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D7·D8·D9·D11
C8	R5·R7·R8·R9·R11·R12·R13·R15·D0·D2·D3·D4·D6·D7·D8·D10
C9	R6·R8·R9·R10·R12·R13·R14·D1·D2·D3·D5·D6·D7·D9
C10	R7·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D8
C11	R8·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D7
C12	R0·R4·R8·R9·R11·D4·D6·D7·D11·D15
C13	R1·R5·R9·R10·R12·D3·D5·D6·D10·D14
C14	R2·R6·R10·R11·R13·D2·D4·D5·D9·D13
C15	R3·R7·R11·R12·R14·D1·D3·D4·D08·D12

The above shows operational expressions in which the input data is made up of 16 bits (D0-D15). If input data length is different from 16 bits, other operational expressions different from the above ones have to be used. For example, in a case where the input data is made up of 8 bits (1 byte: D0-D07), an output data from each flip-flop at a time point when D7 has been input (that is, eight shifts) becomes each of the required operational expressions.

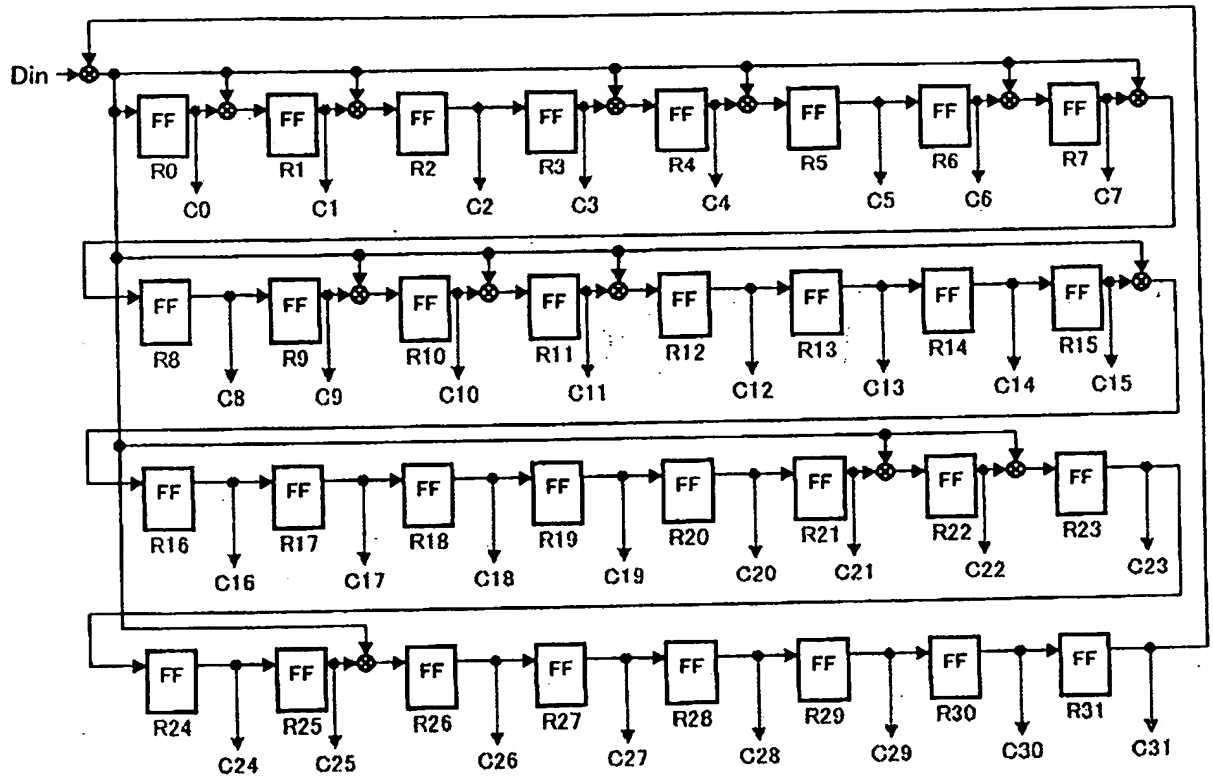
【CRC32】

Generative polynomial

CRC32(32bit) : $G(X) = X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1$

Generating circuit

⊗ : Exclusive-OR



Operational expressions

C0	R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31
C1	R0·R1·R6·R7·R9·R11·R12·R13·R16·R17·R24·R27·R28·D3·D4·D7·D14·D15·D18·D19·D20·D22·D24·D25·D30·D31
C2	R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·D31
C3	R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30
C4	R0·R2·R3·R4·R6·R8·R11·R12·R15·R18·R19·R20·R24·R25·R29·R30·R31·D0·D1·D2·D6·D7·D11·D12·D13·D16·D19·D20·D23·D25·D27·D28·D29·D31
C5	R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31
C6	R1·R2·R4·R5·R6·R7·R8·R11·R20·R21·R25·R30·D1·D2·D6·D9·D10·D11·D17·D20·D23·D24·D25·D26·D27·D29·D30
C7	R0·R2·R3·R5·R7·R8·R10·R15·R16·R21·R22·R23·R24·R28·R29·D2·D3·D6·D7·D8·D9·D10·D15·D16·D21·D23·D24·D26·D28·D29·D31
C8	R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31
C9	R1·R2·R4·R5·R9·R11·R12·R13·R18·R23·R24·R29·D2·D7·D8·D13·D18·D19·D20·D22·D26·D27·D29·D30
C10	R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31
C11	R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·R26·R27·R28·R31·D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·D31
C12	R0·R1·R2·R4·R5·R6·R9·R12·R13·R15·R17·R18·R24·R30·R31·D0·D1·D4·D7·D10·D13·D14·D16·D18·D19·D22·D25·D26·D27·D29·D30·D31
C13	R1·R2·R3·R5·R6·R7·R10·R13·R16·R19·R22·R28·R31·D0·D3·D6·D9·D12·D13·D15·D17·D18·D21·D24·D25·D26·D28·D29·D30
C14	R02 R03 R04 R06 R07 R08 R11 R14 R15 R17 R19 R20 R23 R26 R29 D02 D05 D08 D11 D12 D14 D16 D17 D20 D23 D24 D25 D27 D28 D29
C15	R3·R4·R5·R7·R8·R9·R12·R15·R16·R18·R20·R21·R24·R27·R30·D1·D4·D7·D10·D11·D13·D15·D16·D19·D22·D23·D24·D26·D27·D28
C16	R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·R30·D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31
C17	R1·R5·R6·R9·R13·R14·R18·R20·R22·R25·R27·R30·R31·D0·D1·D4·D6·D8·D9·D11·D13·D17·D18·D22·D25·D26·D30
C18	R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29
C19	R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28
C20	R4·R8·R9·R12·R16·R17·R21·R23·R25·R26·R28·R30·D1·D3·D5·D6·D8·D10·D14·D15·D19·D22·D23·D27
C21	R5·R9·R10·R13·R17·R18·R22·R24·R26·R27·R29·R31·D0·D2·D4·D5·D7·D9·D13·D14·D18·D21·D22·D26
C22	R0·R9·R11·R12·R14·R16·R18·R19·R23·R24·R26·R27·R29·R31·D0·D2·D4·D5·D7·D8·D12·D13·D15·D17·D19·D20·D22·D31

C23	R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·R31·D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31
C24	R1·R2·R7·R10·R14·R16·R17·R18·R20·R21·R27·R28·R30·D1·D3·D4·D10·D11·D13·D14·D15·D17·D21·D24·D29·D30
C25	R2·R3·R8·R11·R15·R17·R18·R19·R21·R22·R28·R29·R31·D0·D2·D3·D9·D10·D12·D13·D14·D16·D20·D23·D28·D29
C26	R0·R3·R4·R6·R10·R18·R19·R20·R22·R23·R24·R25·R26·R28·R31·D0·D3·D5·D6·D7·D8·D9·D11·D12·D13·D21·D25·D27·D28·D31
C27	R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·R29·D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30
C28	R2·R5·R6·R8·R12·R20·R21·R22·R24·R25·R26·R27·R28·R30·D1·D3·D4·D5·D6·D7·D9·D10·D11·D19·D23·D25·D26·D29
C29	R3·R6·R7·R9·R13·R21·R22·R23·R25·R26·R27·R28·R29·R31·D0·D2·D3·D4·D5·D6·D8·D9·D10·D18·D22·D24·D25·D28
C30	R4·R7·R8·R10·R14·R22·R23·R24·R26·R27·R28·R29·R30·D1·D2·D3·D4·D5·D7·D8·D9·D17·D21·D23·D24·D27
C31	R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26

The above shows operational expressions in which the input data is made up of 32 bits (4 bytes: D0-D31). If input data length is different from 32 bits, other operational expressions different from the above ones have to be used. For example, in a case where the input data is made up of 8 bits (1 byte: D0-D07), an output data from each flip-flop at a time point when D7 has been input (that is, eight shifts) becomes each of the required operational expressions. Also, in a case where the input data is made up of 64 bits (8 byte: D0-D63), an output data from each flip-flop at a time point when D63 has been input (that is, 64 shifts) becomes each of the required operational expressions.

The above is all of general explanation about the CRC method and operational expression.

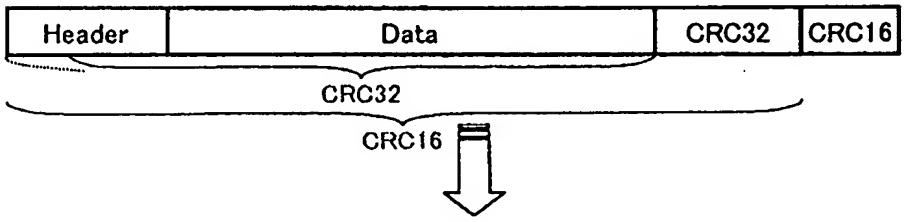
Next, a supplementary explanation relating to the present invention will be given.

【Data Format】

To begin with, a general data format will be explained.

The communications data is generally made up of header information, data, and a CRC arranged before CRC32 in the data format, the CRC16 arithmetic, as described below. The CRC uses generally either of 32 bits and 16 bits in accordance with a requirement for a system (Error detecting methods other than CRC will be omitted from this explanation, since they are not relevant to configurations of the present invention).

The present invention is preferably used in a system (such as an InfiniBand or a like) in which two or more CRC arithmetic operation results are required.



Example: 4-byte (32-bit) transmission

1byte

#1	data0	data1	data2	data3
#2	data4	data5	data6	data7
#3	data8	data9	data10	data11
	⋮			
#n-2	data n-5	data n-4	data n-3	data n-2
#n-1	data n-1	data n	CRC32	CRC32
#n	CRC32	CRC32	CRC16	CRC16

The CRC method is one of error detection methods for detecting error on the data to be transmitted. That is, in this CRC method, when the above-mentioned data are transmitted, it is usable for error detection on data 0 to data n.

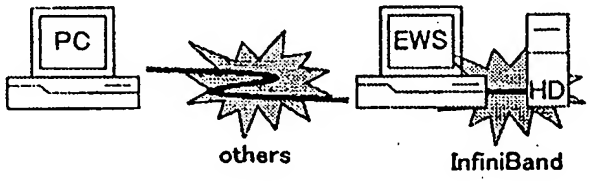
In processing of a CRC16 arithmetic operation, a CRC32 arithmetic operation result is treated as data, and as a result, error detection on data including a CRC32 arithmetic operation result is performed.

For this reason, the CRC16 arithmetic operation needs the CRC32 arithmetic operation result, whereas a CRC32 arithmetic operation does not need a CRC16 arithmetic operation result.

To put it simply, in a data transmitting/receiving, the difference is only whether the CRC32 arithmetic operation result is transmitted earlier or later than the CRC16 arithmetic operation result. If the CRC16 arithmetic operation result is

arranged before the CRC32 arithmetic operation result in the data format, the CRC16 arithmetic operation result is to be used in CRC32 arithmetic operation.

Note: a reason for adding two kinds of CRC code to data will be explained with reference to an example of InfiniBand.



With a system configuration as shown in the above figure (a conventional protocol such as TCP-IP is used as a communications protocol for data communications carried out between a PC (Personal Computer) and a server (EWS), whereas an InfiniBand protocol is used as a communications protocol for data communications carried out between the server (EWS) and an HD (Hard Disk)),

When data access (reading) is made from the PC to HD through a server (EWS), data read from the HD and configured in the foregoing data format is transmitted to the EWS in accordance with the InfiniBand protocol. Next, when the EWS has received data from the HD, the EWS transmits the received data to the PC, without performing a further CRC arithmetic operation. That is, data obtained by removing a header prepared specifically for the InfiniBand protocol and the CRC16 arithmetic operation result from the received data is transmitted from the EWS to the PC. Then, if the CRC32 arithmetic operation result is not added to the received data, the CRC32 arithmetic operation has to be preformed. This means savings in time and power.

【Background of the Invention】

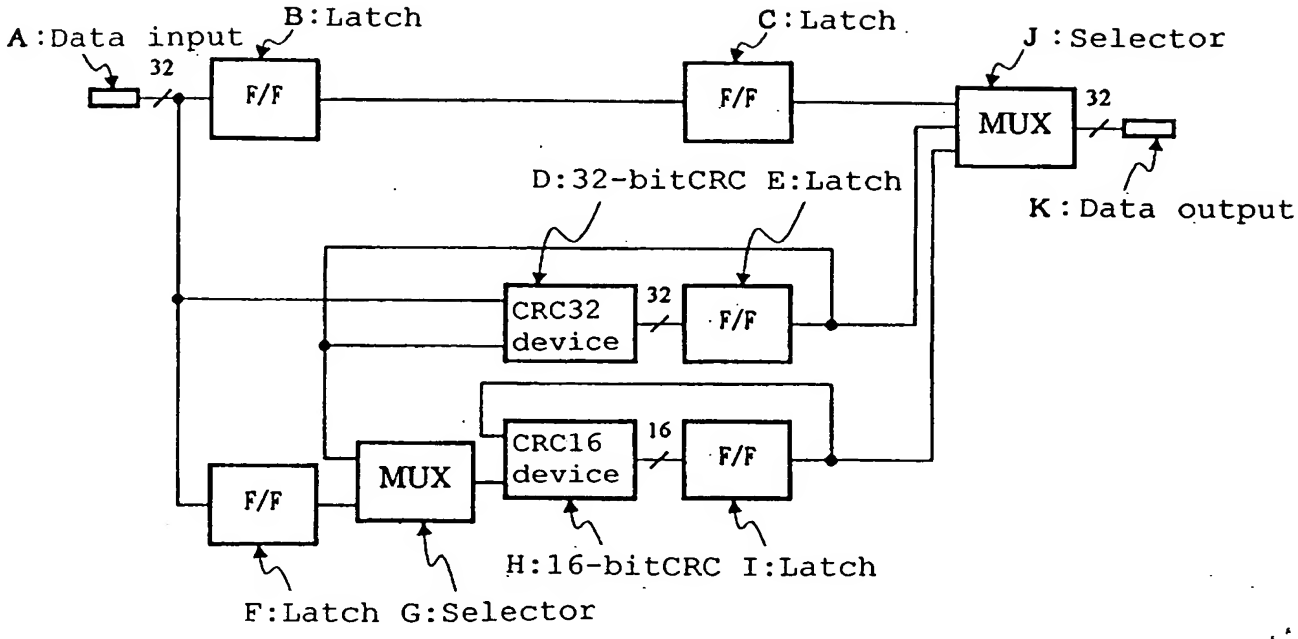
As described above, it is necessary to add the CRC arithmetic operation result to an end of the data to be transmitted.

Generally, in data communications, in order to transmit data accurately to a receiver, continuous transmission from a beginning to an end of the data transmission (in the case of a packet communication, during transmission of one packet) is required. Therefore, a time lag (interruption) between the end of the data and the CRC arithmetic operation result should be avoided. Moreover, to obtain the CRC arithmetic operation result, time being equivalent to at least one clock is necessary.

For this reason, the CRC arithmetic operation result has to follow continuously the end of the data to be transmitted, by inserting latches between Data paths. Furthermore, in a case of having two kinds of CRC arithmetic operation results, at least two latches have to be inserted between Data paths, since time being equivalent to at least two clocks is necessary (in order to use one CRC arithmetic operation result in another arithmetic operation).

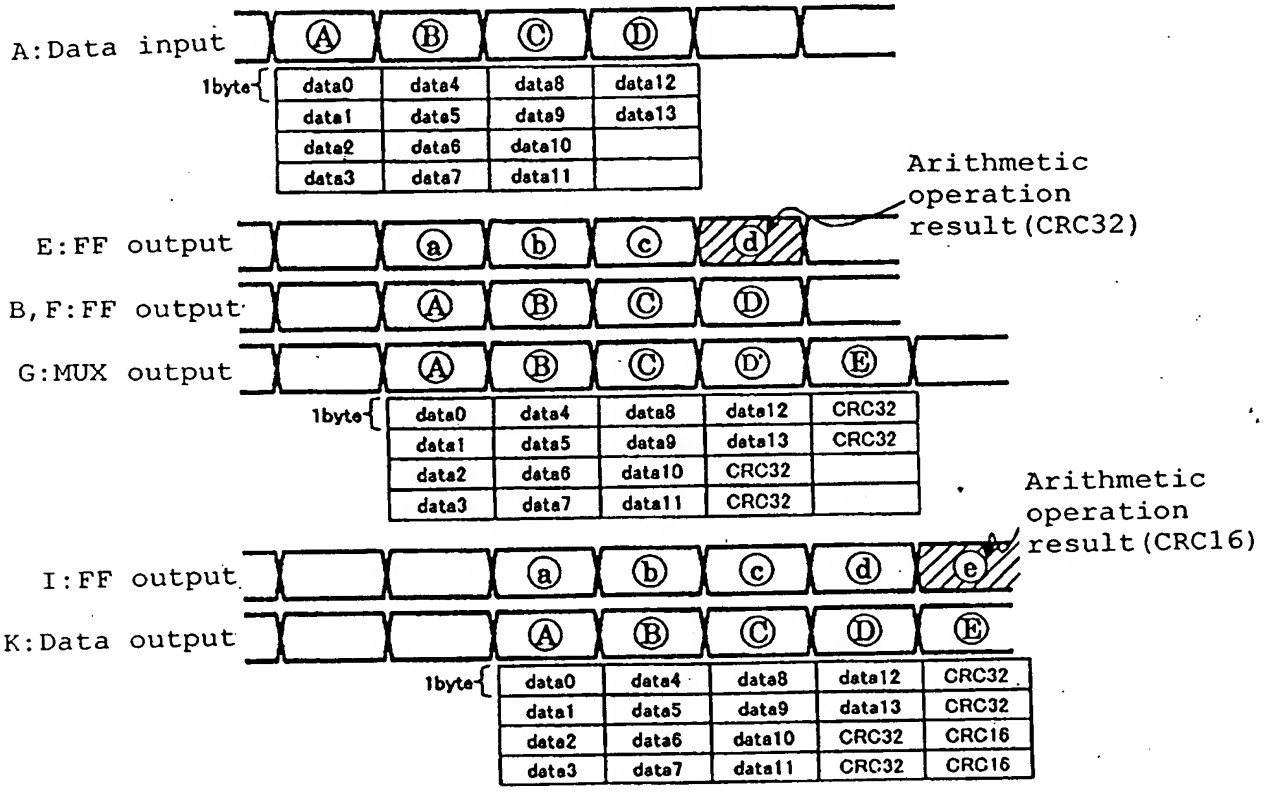
High-speed signal processing also in data communications has become indispensable, as high-speed operations of CPUs (Central Processing Units) have been achieved in recent years. To achieve high-speed signal processing, it may be preferable to increase a data transmission speed and/or a width of a bus. In addition to these, an increase in the processing speed within a signal processing circuit is essential.

【Conventional Circuit Configuration】



A: Data inputting section
B, C: Latch (32-bit flip-flop) for adjusting operation timing in data path
D: CRC32 arithmetic operation device
E: Latch (32-bit flip-flop) for latching CRC32 arithmetic operation result
F: Latch (16-bit flip-flop) for adjusting operation timing of CRC16 arithmetic operation device
G: Selector circuit for selecting either of latched input data and CRC32 arithmetic operation result
H: CRC16 arithmetic operation device
I: Latch (16-bit flip-flop) for latching CRC16 arithmetic operation result
J: Output selector
K: Data outputting section

Thirty-two bits (4 bytes) of data output from a data inputting section A are input to a latch B, a CRC32 arithmetic operation device D, and a latch F which is used to adjust operation timing of a CRC16 arithmetic operation device. A CRC32 arithmetic operation result (an arithmetic operation result of 32 bits output from the CRC32 arithmetic operation device D) is input to an output selector J, and a selector circuit G through a latch E. In the selector circuit G, either of the input data and the CRC32 arithmetic operation result is selected as data to be input to a CRC16 arithmetic operation device H. A CRC16 arithmetic operation result (an arithmetic operation result of 16 bits output from the CRC16 arithmetic operation device H) is input to the output selector J through a latch I.



Let it be assumed that data as shown in the above timing chart is input from the data inputting section A, and that an end part (D) of the input data is made up of only 2 bytes (16 bits). The first CRC32 arithmetic operation is performed by using a first part (A) of the input data and the initial value of the latch E. The latch E latches the first arithmetic operation result obtained from the CRC32 arithmetic operation device D. After this, the second CRC32 arithmetic operation is performed by using a second part (B) of the input data and the first CRC32 arithmetic

operation result (data latched in the latch E). By repeating the above CRC32 arithmetic operations, the CRC32 code bit ⓓ can finally be obtained.

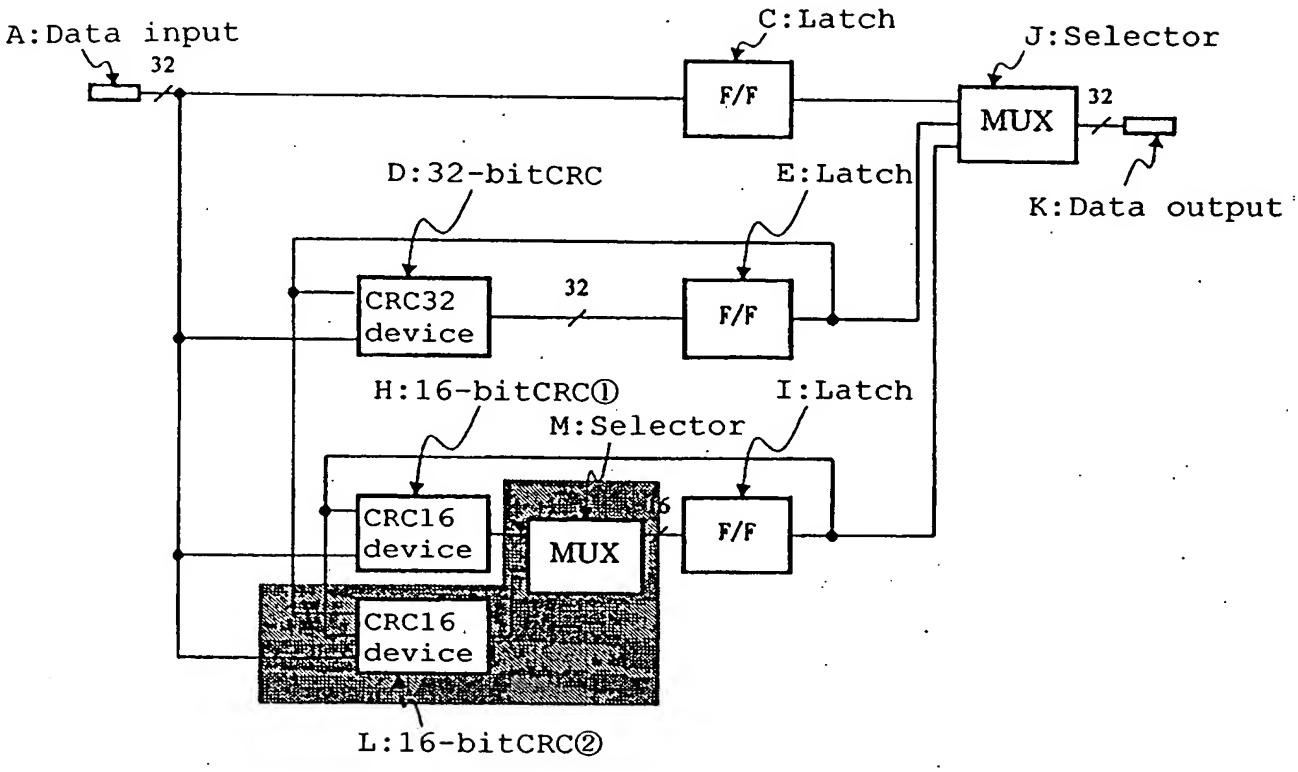
Next, it is necessary to add continuously the CRC32 arithmetic operation result ⓓ to the end part Ⓓ of the input data fed from the data inputting section A and then to feed the added data to the CRC16 arithmetic operation device H.

Then, in a case where the end part Ⓓ of the input data is made up of only 2 bytes, it is necessary to input the CRC32 arithmetic operation result separately at two timings, as shown in the above timing chart. For this reason, it is necessary to delay the input data by one clock. Therefore, the latch F is provided on an input side of selector circuit (MUX section) G.

Thus, the CRC16 code bit ⓔ can finally be obtained.

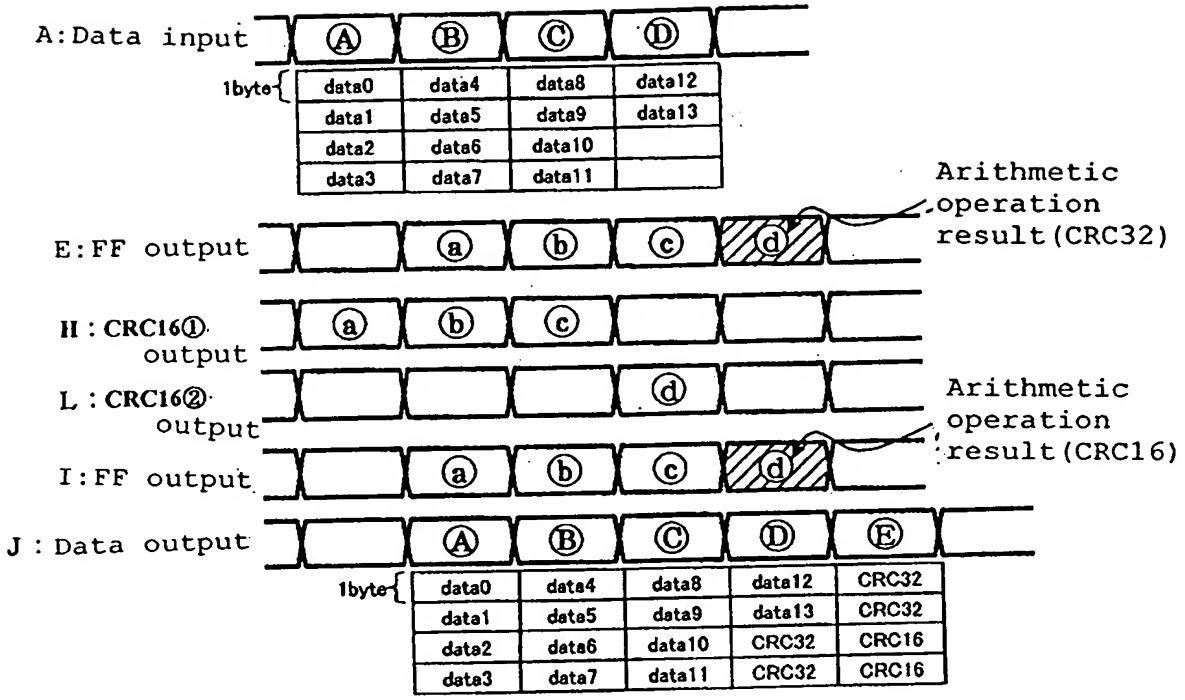
With the conventional circuit configuration described above, a time delay being equivalent to two clocks occurs between inputting of data and outputting of data.

【Circuit Configuration of the Invention】



A: Data inputting section
C: Latch (32-bit flip-flop) for adjusting operation timing in data path
D: CRC32 arithmetic operation device
E: Latch (32-bit flip-flop) for latching CRC32 arithmetic operation result
H: CRC16 arithmetic operation device ①
I: Latch (16-bit flip-flop) for latching CRC16 arithmetic operation result
J: Output selector
K: Data outputting section
L: CRC16 arithmetic operation device ② (for reducing latency)
M: Output selector

Thirty-two bits (4 bytes) of data output from a data inputting section A are input to a latch C, a CRC32 arithmetic operation device D, a CRC16 arithmetic operation device ① H, and a CRC16 arithmetic operation device ② L. Arithmetic operation results obtained from these arithmetic operation devices are input to the output selector J through respective latches E and I.



Let it be assumed that data as shown in the above timing chart is input from the data inputting section A (same as the conventional example).

The first CRC32 arithmetic operation is performed by using the first part (A) of the input data and the initial value of the latch E. The latch E latches the first arithmetic operation result obtained from the CRC32 arithmetic operation device D. After this, the second CRC32 arithmetic operation is performed by using the second part (B) of the input data and the first CRC32 arithmetic operation result(data latched in the latch E). By repeating the above CRC32 arithmetic operations, the CRC32 code bit (d) can finally be obtained.

Next, the CRC16 arithmetic operation device ① H performs the first CRC16 arithmetic operation by using the first part (A) of the input data and the initial value of the latch I. The latch I latches the first arithmetic operation result obtained from a CRC16 arithmetic operation device ① H. After this, the second CRC32 arithmetic operation is performed by using the second part (B) of the input data and the first CRC16 arithmetic operation result(data latched in the latch I).

The above CRC16 arithmetic operations are repeated up to the data part (C), immediately (one clock) before the end part (D) of the input data, the CRC32 code bit (d) can finally be obtained.

When the end part (D) of the input data is detected, an output selector (MUX) M selects the CRC16 arithmetic operation device ② L, and the CRC16 code bit (d) can finally be obtained from the CRC16 arithmetic operation device ② L.

The CRC16 arithmetic operation device ② L inputs the end part (D) of the input data, the CRC16 ① arithmetic operation result (c) obtained from the CRC16 arithmetic operation device ① H through the latch I, and the CRC32 arithmetic operation result (c) obtained from the CRC32 arithmetic operation device D through the latch E, in order to expedite timing.

At this stage, the CRC32 arithmetic operation result is obtained by performing the CRC32 arithmetic operation using the end part (D) of the input data and the immediately preceding CRC32 arithmetic operation result.

Accordingly, with the above configuration of the present invention having feature in that the CRC32 arithmetic operation is included (incorporated) in the CRC16 arithmetic operation, it is possible to perform the CRC16 arithmetic operation, without using (waiting for) the CRC32 arithmetic operation result.

The CRC16 arithmetic operation result can be obtained one clock (MIN.) earlier, compared to that in the conventional configuration, since it is not necessary to wait for the CRC32 arithmetic operation result. Only one clock delay occurs even on the side of the data path.

With the circuit configuration of the present invention, a time delay being equivalent to one clock occurs between inputting of data and outputting of data. This means reduction of latency by one clock (MIN.), compared to the conventional circuit configuration.

【Operational Expression Producing Method】

The CRC arithmetic operation devices, which are used in a conventional example and an embodiment according to the present invention, use the following operational expression.

CRC32 arithmetic operation device D:

The device D uses the operational expression described on pages 6 to 8 of this Document.

CRC16 arithmetic operation device H:

The device H utilizes the generating circuit described on page 5 and uses a set of output data being output from each of the flip-flops when 32 bits of data were shifted. In fact, the operational expression described on page 5 is an example in a case where 16 bits of data are shifted.

CRC16 arithmetic operation device L:

The device L produces newly an operational expression by using the following method:

As described under the section "circuit configuration of the Invention", if the CRC16 arithmetic operation is performed after the CRC32 arithmetic operation result was obtained, a time delay being equivalent to one clock occurs inevitably.

To solve this problem, it is preferable that the CRC32 arithmetic operation is simultaneously performed, when the CRC16 arithmetic operation is performed, whereby it becomes possible to acquire simultaneously the CRC32 and the CRC16 arithmetic operation results, without occurrence of a time delay.

Therefore, in order to avoid such a time delay, a new operational expression is produced and used according to the procedures as below:

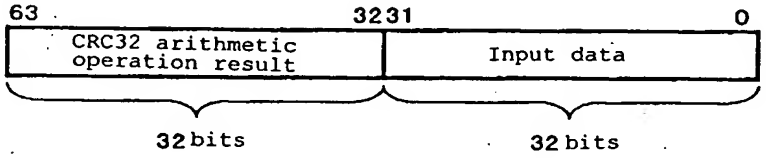
① data being of 64 bits in length.

In the CRC16 arithmetic operation device, the operational expression is produced using input data (32 bits) and an immediately preceding arithmetic operation result (16 bits).

With the conventional method, CRC16 code bit is acquired by adding CRC32 code bit (32 bits) to an end part of the input data.

The CRC16 arithmetic operation device ② L incorporated in the present invention inputs simultaneously the input data and CRC32 arithmetic operation result (obtained immediately before a final CRC32 arithmetic operation result). That is, the operational expression is produced as 64 bits of the input data. At this stage, original input data (D) as lower-order bits and

the CRC arithmetic operation result as higher-order bits are respectively input.



②Operational Expression Production-I

First, a CRC16 operational expression on input data being of 64 bits in length is produced.

This is the output data from each of the flip-flops making up the CRC16 generating circuit described on page 5, when 64 bits of data is shifted.

C0	R2·R4·R5·R8·R9·R11·R12·R13·R14·D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63
C1	R2·R3·R4·R6·R8·R10·R11·R15·D0·D4·D5·D7·D9·D11·D12·D13·D16·D20·D21·D22·D25·D26·D28·D29·D30·D32·D33·D35·D36·D37·D39·D40·D41·D43·D47·D48·D49·D51·D54·D55·D58·D59·D62·D63
C2	R0·R3·R4·R5·R7·R9·R11·R12·D3·D4·D6·D8·D10·D11·D12·D15·D19·D20·D21·D24·D25·D27·D28·D29·D31·D32·D34·D35·D36·D38·D39·D40·D42·D46·D47·D48·D50·D53·D54·D57·D58·D61·D62
C3	R1·R2·R6·R9·R10·R11·R14·D1·D4·D5·D6·D9·D13·D14·D17·D22·D23·D24·D27·D28·D29·D30·D32·D33·D38·D39·D40·D41·D42·D43·D45·D46·D47·D48·D49·D50·D51·D52·D53·D55·D56·D57·D59·D60·D61·D63

C4	R2·R3·R7·R10·R11·R12·R15·D0·D3·D4·D5·D8·D12·D13·D16·D21·D22· D23·D26·D27·D28·D29·D31·D32·D37·D38·D39·D40·D41·D42·D44·D45· D46·D47·D48·D49·D50·D51·D52·D54·D55·D56·D58·D59·D60·D62
C5	R0·R3·R4·R8·R11·R12·R13·D02·D03·D4·D7·D11·D12·D15·D20·D21·D22· D25·D26·D27·D28·D30·D31·D36·D37·D38·D39·D40·D41·D43·D44·D45· D46·D47·D48·D49·D50·D51·D53·D54·D55·D57·D58·D59·D61
C6	R1·R4·R5·R9·R12·R13·R14·D1·D2·D3·D6·D10·D11·D14·D19·D20·D21· D24·D25·D26·D27·D29·D30·D35·D36·D37·D38·D39·D40·D42·D43·D44· D45·D46·D47·D48·D49·D50·D52·D53·D54·D56·D57·D58·D60
C7	R2·R5·R6·R10·R13·R14·R15·D0·D1·D2·D5·D09·D10·D13·D18·D19·D20· D23·D24·D25·D26·D28·D29·D34·D35·D36·D37·D38·D39·D41·D42·D43· D44·D45·D46·D47·D48·D49·D51·D52·D53·D55·D56·D57·D59
C8	R3·R6·R7·R11·R14·R15·D0·D1·D4·08·D9·D12·D17·D18·D19·D22·D23· D24·D25·D27·D28·D33·D34·D35·D36·D37·D38·D40·D41·D42·D43·D44· D45·D46·D47·D48·D50·D51·D52·D54·D55·D56·D58
C9	R4·R7·R8·R12·R15·D0·D3·D7·D8·D11·D16·D17·D18·D21·D22·D23·D24· D26·D27·D32·D33·D34·D35·D36·D37·D39·D40·D41·D42·D43·D44·D45· D46·D47·D49·D50·D51·D53·D54·D55·D57
C10	R0·R5·R08·R9·R13·D2·D6·D7·D10·D15·D16·D17·D20·D21·D22·D23·D25· D26·D31·D32·D33·D34·D35·D36·D38·D39·D40·D41·D42·D43·D44·D45· D46·D48·D49·D50·D52·D53·D54·D56
C11	R0·R1·R6·R9·R10·R14·D1·D5·D6·D9·D14·D15·D16·D19·D20·D21·D22· D24·D25·D30·D31·D32·D33·D34·D35·D37·D38·D39·D40·D41·D42·D43· D44·D45·D47·D48·D49·D51·D52·D53·D55
C12	R0·R1·R4·R5·R7·R8·R9·R10·R12·R13·R14·R15·D0·D1·D2·D3·D5·D6·D7· D8·D10·D11·D14·D15·D17·D21·D22·D23·D24·D26·D30·D33·D35·D36· D38·D39·D41·D44·D46·D47·D52·D54·D55·D59·D63
C13	R1·R2·R5·R6·R8·R9·R10·R11·R13·R14·R15·D0·D1·D2·D4·D5·D6·D7·D9· D10·D13·D14·D16·D20·D21·D22·D23·D25·D29·D32·D34·D35·D37·D38· D40·D43·D45·D46·D51·D53·D54·D58·D62
C14	R0·R2·R3·R6·R7·R9·R10·R11·R12·R14·R15·D0·D1·D3·D4·D5·D6·D8·D9· D12·D13·D15·D19·D20·D21·D22·D24·D28·D31·D33·D34·D36·D37·D39· D42·D44·D45·D50·D52·D53·D57·D61
C15	R1·R3·R4·R7·R8·R10·R11·R12·R13·R15·D0·D2·D3·D4·D5·D7·D8·D11· D12·D14·D18·D19·D20·D21·D23·D27·D30·D32·D33·D35·D36·D38·D41· D43·D44·D49·D51·D52·D56·D60

③. Replacement of data

Operational expressions described on pages 8-9 are substituted into operational expressions obtained in "②", since D63-D31 are the CRC32 arithmetic operation results, as clear from "①".

Substituting

C0= R2·R4·R5·R8·R9·R11·R12·R13·R14·
D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·
D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26
R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·R29·
D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30
R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·R31·
D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31
R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·
D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28
R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·
D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29
R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·R30·
D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31
R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·R26·R27·R28·R31·
D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·D31
R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·
D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31
R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·
D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31
R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·
D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31
R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·
D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30
R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·
D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·D31
R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·
D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31

Deleting same terms

C0= Z2·Z4·Z5·Z8·Z9·Z11·Z12·Z13·Z14·
R1·R4·R5·R6·R9·R10·R13·R14·R19·R20·R22·R24·R28·R31·
D0·D1·D2·D4·D6·D9·D10·D12·D13·D19·D20·D21·D25·D27·D29·D30·D31·

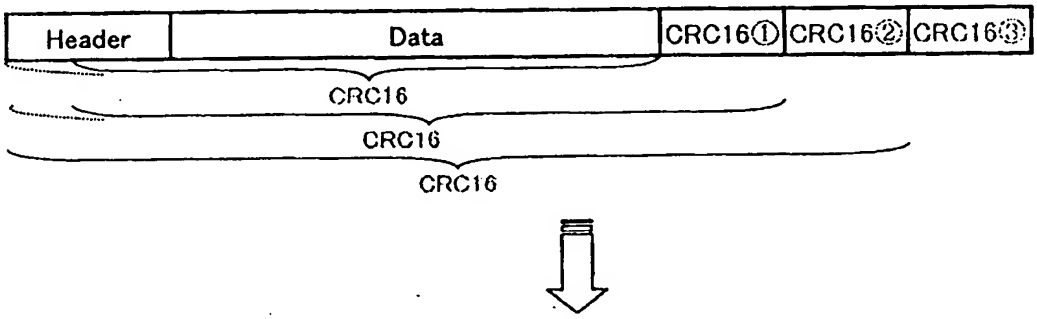
It is possible to obtain simultaneously final CRC32 and CRC16 arithmetic operation results, by performing the above processing on all the operational expressions.

Three CRC16 arithmetic operation devices are incorporated in a circuit of a second embodiment of the present invention. A data format, a circuit configuration, operation timing and operational expressions in the second embodiment will be described below.

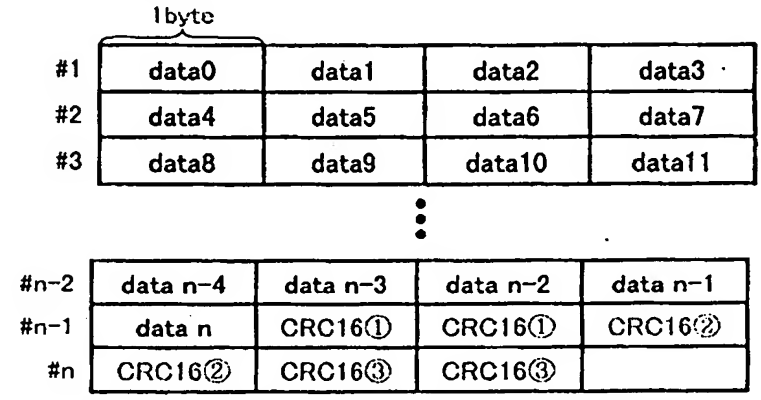
【Data Format】

To begin with, a data format in a case where three CRC16 arithmetic operation devices are required will be explained.

Please note that, even if another arithmetic operation (CRC32 arithmetic operation device) being different from the CRC16 arithmetic operation is used, new operational expressions can be produced using a same circuit configuration in accordance with an algorithm described below.

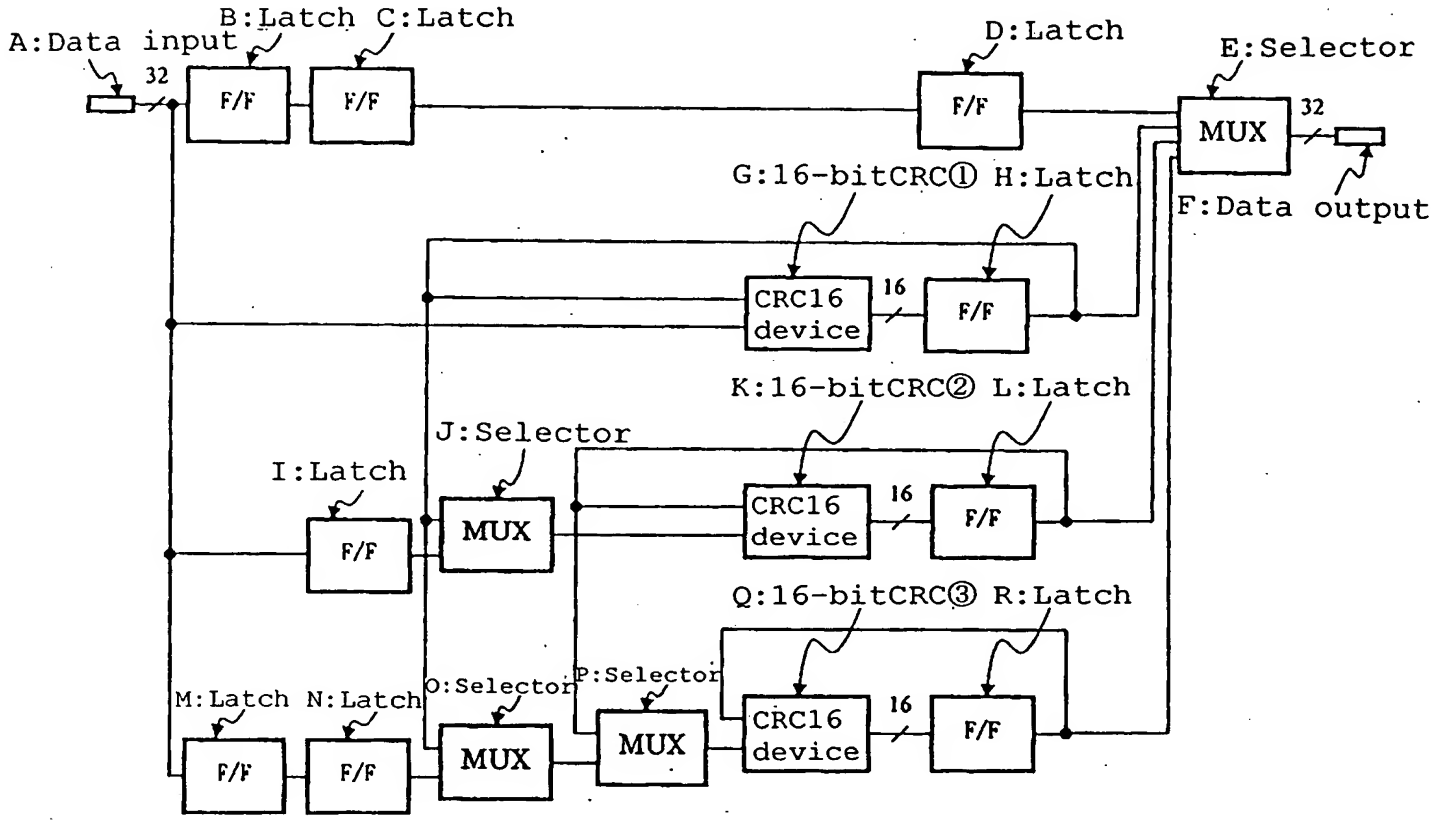


Example: 4-byte (32-bit) transmission

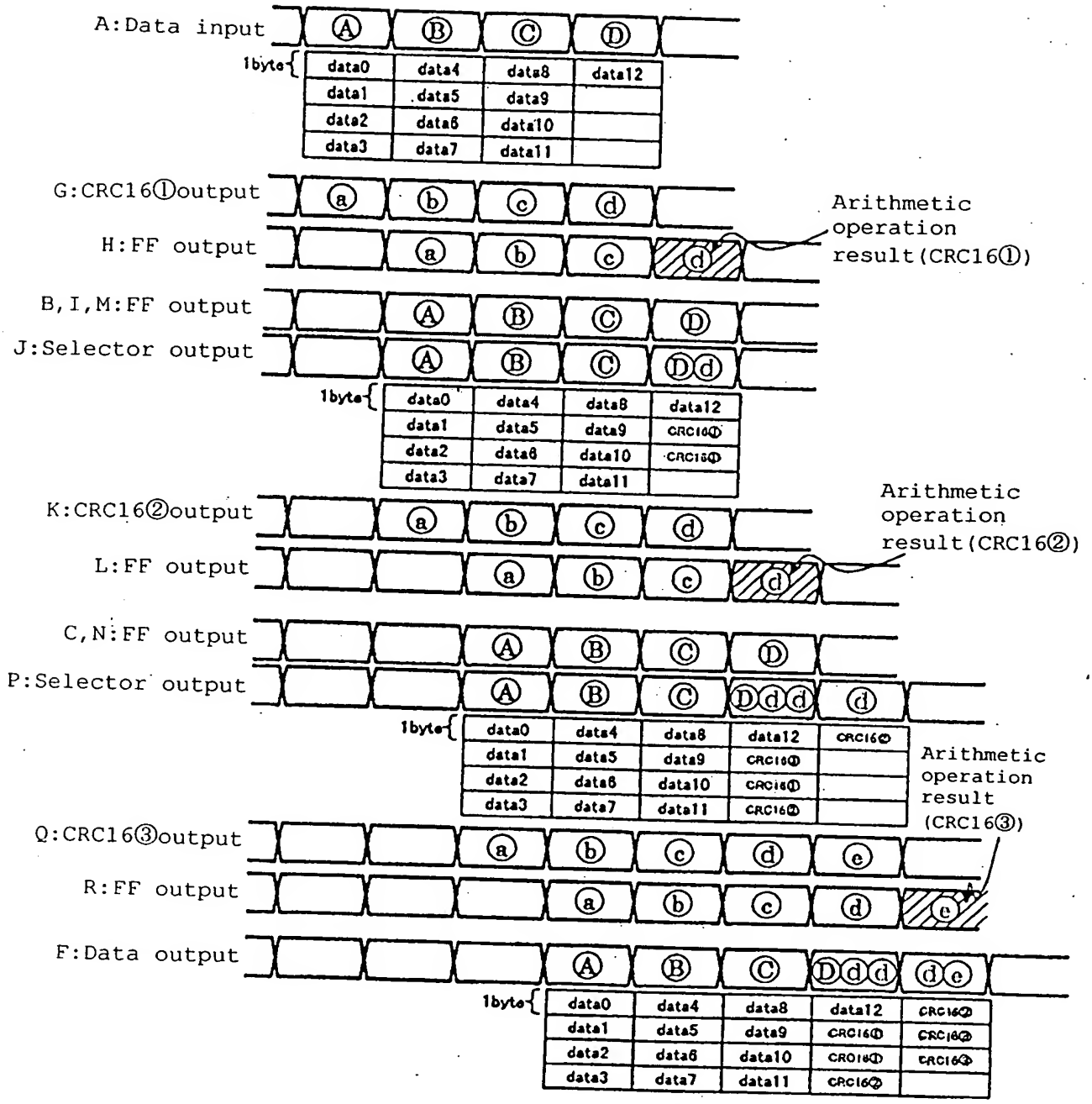


In processing of a CRC16 ② arithmetic operation, a CRC16 ① arithmetic operation result is treated as data, and as a result, error detection on data including a CRC16 ① arithmetic operation result is performed. In addition, in processing of a CRC16 ③ arithmetic operation, the CRC16 ① and CRC16 ② arithmetic operation results are treated as data, and as a result, error detection on data including the CRC16 ② arithmetic operation result is performed.

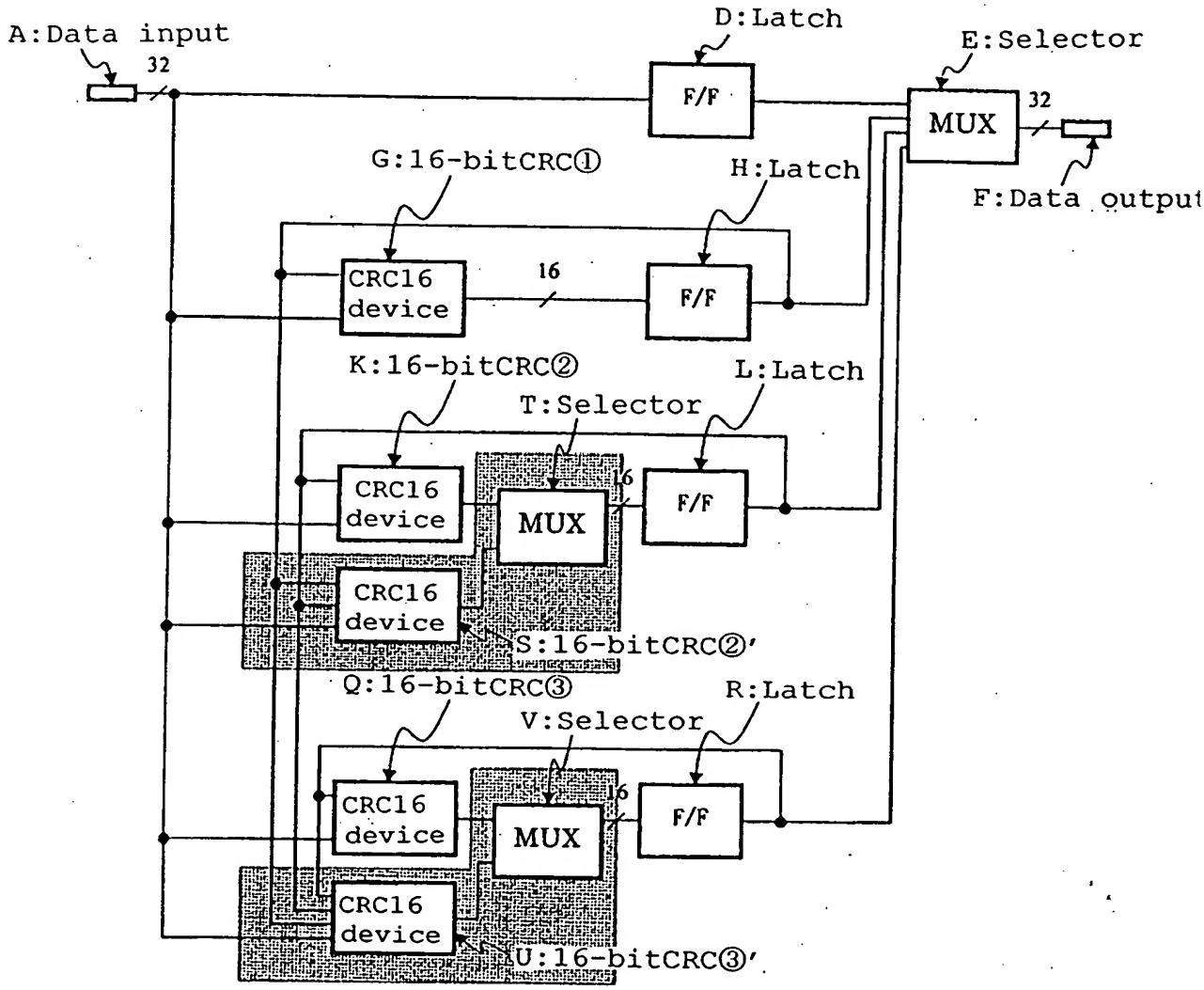
【Circuit Configuration according to the Conventional Method】



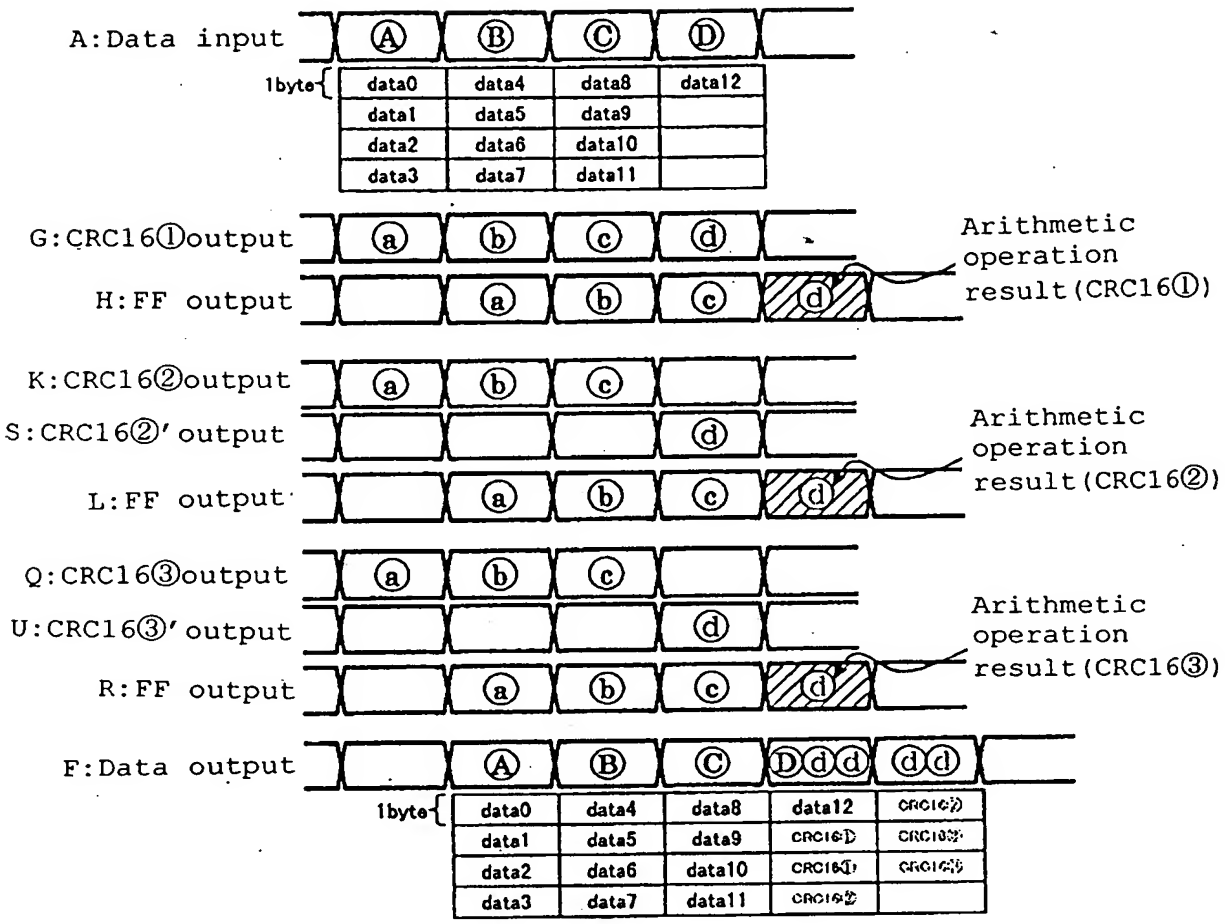
A: Data inputting section
B, C, D: Latch (32-bit flip-flop) for adjusting operation timing in data path
E: Output selector
F: Data outputting section
G: CRC16 arithmetic operation device ①
H: Latch (16-bit flip-flop) for latching CRC16 ① arithmetic operation result
I: Latch (32-bit flip-flop) for adjusting operation timing of CRC16 ② arithmetic operation device
J, O: Selector circuit for selecting either of latched input data and CRC16 ① arithmetic operation result
K: CRC16 arithmetic operation device ② (for reducing latency)
L: Latch (16-bit flip-flop) for latching CRC16 ② arithmetic operation result
M, N: Latch (32-bit flip-flop) for adjusting operation timing of CRC16 ② arithmetic operation device
P: Selector circuit for selecting either of output from selector circuit O and CRC16 ② arithmetic operation result
Q: CRC16 arithmetic operation device ③
R: Latch (16-bit flip-flop) for latching CRC16 ③ arithmetic operation result


A:Data input
1byte
data0 data4 data8 data12
data1 data5 data9
data2 data6 data10
data3 data7 data11
G:CRC16①output
H:FF output
Arithmetic operation result(CRC16①)
B,I,M:FF output
J:Selector output
1byte
data0 data4 data8 data12
data1 data5 data9 CRC16①
data2 data6 data10 CRC16①
data3 data7 data11
K:CRC16②output
L:FF output
Arithmetic operation result(CRC16②)
C,N:FF output
P:Selector output
1byte
data0 data4 data8 data12 CRC16②
data1 data5 data9 CRC16①
data2 data6 data10 CRC16①
data3 data7 data11 CRC16①
Arithmetic operation result (CRC16③)
Q:CRC16③output
R:FF output
F:Data output
1byte
data0 data4 data8 data12 CRC16②
data1 data5 data9 CRC16① CRC16②
data2 data6 data10 CRC16① CRC16②
data3 data7 data11 CRC16②

【Circuit Configuration according to Second Embodiment】



A: Data inputting section
D: Latch (32-bit flip-flop) for adjusting operation timing in data path
E: Output selector
F: Data outputting section
G: CRC16 arithmetic operation device ①
H: Latch (16-bit flip-flop) for latching CRC16 ① arithmetic operation result
K: CRC16 arithmetic operation device ②
L: Latch (16-bit flip-flop) for latching CRC16 ② arithmetic operation result
Q: CRC16 arithmetic operation device ③
R: Latch (16-bit flip-flop) for latching CRC16 ③ arithmetic operation result
S: CRC16 arithmetic operation device ②' (for reducing latency)
T: Output selector for selecting either of CRC16 arithmetic operation devices ②, ②'
U: CRC16 arithmetic operation device ③' (for reducing latency)
V: Output selector for selecting either of CRC16 arithmetic operation devices ③, ③'



Let it be assumed that data as shown in the above timing chart is input from the data inputting section A (same as the conventional example).

The first CRC16 ① arithmetic operation is performed by using the first part Ⓐ of the input data and the initial value of the latch H. The latch H latches the first arithmetic operation result obtained from the CRC16 ① arithmetic operation device G. After this, the second CRC16 ① arithmetic operation is performed by using the second part Ⓑ of the input data and the first CRC16 ① arithmetic operation result (data latched in the latch H). By repeating the above CRC16 ① arithmetic operations, the CRC16 ① code bit ⓓ can finally be obtained.

Next, in a same manner as described above, the CRC16 arithmetic operation device ② K also performs the first CRC16 ② arithmetic operation by using the first part Ⓐ of the input data and the initial value of the latch L.

The latch L latches the first arithmetic operation result obtained from a CRC16 arithmetic operation device ② K. The above

CRC16 ② arithmetic operations are repeated up to the data part ⓒ immediately (one clock) before the end part ⓓ of the input data, the CRC32 code bit · can finally be obtained. When the end part ⓓ of the input data is detected, an output selector (MUX) T selects the CRC16 arithmetic operation device ②' S, and the CRC16 ② code bit ⓐ can finally be obtained from the CRC16 arithmetic operation device ②' S. The CRC16 arithmetic operation device ②' S inputs the end part ⓓ of the input data, the CRC16 ② arithmetic operation result ⓒ obtained from the CRC16 arithmetic operation device ② K through the latch L, and the CRC16 ① arithmetic operation result ⓒ obtained from the CRC16 ① arithmetic operation device D through the latch H, in order to expedite timing.

At this stage, the CRC16 ① arithmetic operation result is obtained by performing the CRC16 ① arithmetic operation using the end part· of the input data and the immediately preceding CRC16 ① arithmetic operation result.

Accordingly, with the above configuration of the second invention having feature in that the CRC16 ① arithmetic operation is included (incorporated) in the CRC16 ② arithmetic operation, it is possible to perform the CRC16 ② arithmetic operation, without using (waiting for) the CRC16 ① arithmetic operation result.

The CRC16 ② arithmetic operation result can be obtained one clock (MIN.) earlier, compared to that in the conventional configuration, since it is not necessary to wait for the CRC16 ① arithmetic operation result. Only one clock delay occurs even on the side of the data path.

In addition, with configuration of the second embodiment, in a stage of the third arithmetic operation performed by the CRC16 arithmetic operation device ③', since the CRC16 ① and CRC16 ② arithmetic operations are included (incorporated) in the CRC16 ③' arithmetic operation, it is possible to perform the CRC16 ② arithmetic operation, without using (waiting for) the CRC16 ① and CRC16 ② arithmetic operation results.

With the circuit configuration of the second embosiment, a time delay being equivalent to one clock occurs between inputting of data and outputting of data. This means reduction of latency by two clocks (MIN.), compared to the conventional circuit configuration.

【operational expression producing method】

The CRC arithmetic operation devices, which are used in a conventional example and an embodiment according to the present invention, use the following operational expression.

CRC16 arithmetic operation device G, K, Q:

The device G, K, Q utilize the generating circuit described on page 5 of the previous Document and use a set of output data being output from each of the flip-flops when 32 bits of data were shifted.

CRC16 arithmetic operation device S:

The device S uses the operational expression described on page 11 of this Document obtained in accordance with the operational expression producing method described in the previous Document.

CRC16 arithmetic operation device U:

The device U produces newly an operational expression by using the following method:

If the CRC16 ② arithmetic operation is performed after the CRC16 ① and CRC16 ② arithmetic operation results arithmetic operation result was obtained, a time delay being equivalent to two clocks occurs inevitably.

To solve this problem, it is preferable that the CRC16 ① arithmetic operation is simultaneously performed, when the CRC16 ② arithmetic operation is performed, and the CRC16 ① and CRC16 ② arithmetic operation is simultaneously performed, when the CRC16 ③ arithmetic operation is performed, whereby it becomes possible to acquire simultaneously the CRC16 ①, CRC16 ② and CRC16 ③ arithmetic operation results, without using (waiting for) the CRC16 ① and CRC16 ② arithmetic operation results.

Therefore, in order to avoid such a time delay, a new operational expression is produced and used according to the procedures as below:

① Operational Expression Production

CRC16 arithmetic operation device ① G; ② K; ③ Q;

The device G, K, Q utilize the generating circuit described in the previous Document, and a set of output data being output from each of the flip-flops, when 32 bits of data were shifted, are as follows:

	operational expression (⊕:ExculsiveOR)
C15	X03·X04·X06·X09·X11·X12·X14·X15·/ D00·D01·D03·D04·D06·D09·D11·D12·D17·D19·D20·D24·D28
C14	X02·X03·X05·X08·X10·X11·X13·X14· D01·D02·D04·D05·D07·D10·D12·D13·D18·D20·D21·D25·D29
C13	X01·X02·X04·X07·X09·X10·X12·X13·X15· D00·D02·D03·D05·D06·D08·D11·D13·D14·D19·D21·D22·D26·D30
C12	X00·X01·X03·X06·X08·X09·X11·X12·X14· D01·D03·D04·D06·D07·D09·D12·D14·D15·D20·D22·D23·D27·D31
C11	X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15· D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·D19·D20·D21· D23
C10	X01·X02·X03·X04·X05·X06·X07·X08·X09·X11·X12·X13·X14·X15· D00·D01·D02·D03·D04·D06·D07·D08·D09·D10·D11·D12·D13·D14·D16·D17·D18·D20·D21· D22·D24
C09	X00·X01·X02·X03·X04·X05·X06·X07·X08·X10·X11·X12·X13·X14·X15· D00·D01·D02·D03·D04·D05·D07·D08·D09·D10·D11·D12·D13·D14·D15·D17·D18·D19·D21· D22·D23·D25
C08	X00·X01·X02·X03·X04·X05·X06·X07·X09·X10·X11·X12·X13·X14· D01·D02·D03·D04·D05·D06·D08·D09·D10·D11·D12·D13·D14·D15·D16·D18·D19·D20·D22· D23·D24·D26
C07	X00·X01·X02·X03·X04·X05·X06·X08·X09·X10·X11·X12·X13· D02·D03·D04·D05·D06·D07·D09·D10·D11·D12·D13·D14·D15·D16·D17·D19·D20·D21·D23· D24·D25·D27
C06	X00·X01·X02·X03·X04·X05·X07·X08·X09·X10·X11·X12· D03·D04·D05·D06·D07·D08·D10·D11·D12·D13·D14·D15·D16·D17·D18·D20·D21·D22·D24· D25·D26·D28
C05	X00·X01·X02·X03·X04·X06·X07·X08·X09·X10·X11· D04·D05·D06·D07·D08·D09·D11·D12·D13·D14·D15·D16·D17·D18·D19·D21·D22·D23·D25· D26·D27·D29
C04	X00·X01·X02·X03·X05·X06·X07·X08·X09·X10·X15· D00·D05·D06·D07·D08·D09·D10·D12·D13·D14·D15·D16·D17·D18·D19·D20·D22·D23·D24· D26·D27·D28·D30
C03	X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15· D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·D20·D21·D23·D24· D25·D27·D28·D29·D31
C02	X00·X01·X05·X07·X08·X09·X11·X12·X13·X15· D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·D26·D29·D30
C01	X00·X04·X06·X07·X08·X10·X11·X12·X14·X15· D00·D01·D03·D04·D05·D07·D08·D09·D11·D15·D16·D17·D19·D22·D23·D26·D27·D30·D31
C00	X04·X05·X07·X10·X12·X13·X15· D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

"X" denotes an initial value of latch (flip-flop) H.

② RC16 arithmetic operation device ②′ S:
Operational Expression Production

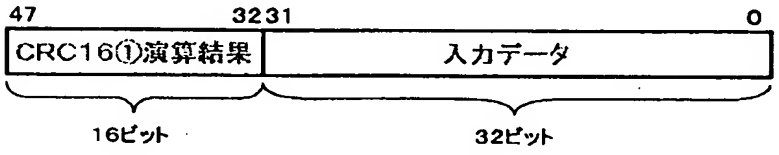
The device S uses the operational expression described in this Document obtained in accordance with the operational expression producing method described in the previous Document.

a. Data being of 48 bits in length.

In the CRC16 arithmetic operation device ②′ S, the operational expression is produced using input data (32 bits) and an immediately preceding arithmetic operation result (16 bits).

With the conventional method, CRC16 code bit is acquired by adding the CRC16 code bit (result obtained from CRC16 arithmetic operation device ①: 16 bits) to an end part of the input data.

The CRC16 arithmetic operation device ②′ S incorporated in the present invention inputs simultaneously the input data, and CRC16 ① arithmetic operation result (obtained one operation before a final CRC16 arithmetic operation result). That is, the operational expression is produced as 48 bits of the input data. At this stage, original input data · as lower-order bits and the CRC arithmetic operation result as higher-order bits (see below) are respectively input.



b. Operational Expression Production-I
First, a CRC16 operational expression on input data being of 48 bits in length is produced. This is the output data from each of the flip-flops making up the CRC16 generating circuit described in the previous document, when 48 bits of data is shifted.

on Page 6

	operational expressions (·:ExculsiveOR)
C15	Z01·Z04·Z08·Z10·Z11·Z12·Z13· D02·D03·D04·D05·D07·D11·D14·D16·D17·D19·D20·D22·D25·D27·D28·D33·D35·D36·D40·D44
C14	Z00·Z03·Z07·Z09·Z10·Z11·Z12· D03·D04·D05·D06·D08·D12·D15·D17·D18·D20·D21·D23·D26·D28·D29·D34·D36·D37·D41·D45
C13	Z02·Z06·Z08·Z09·Z10·Z11·Z15· D00·D04·D05·D06·D07·D09·D13·D16·D18·D19·D21·D22·D24·D27·D29·D30·D35·D37·D38·D42·D46
C12	Z01·Z05·Z07·Z08·Z09·Z10·Z14· D01·D05·D06·D07·D08·D10·D14·D17·D19·D20·D22·D23·D25·D28·D30·D31·D36·D38·D39·D43·D47
C11	Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15· D00·D03·D04·D05·D06·D08·D09·D14·D15·D16·D17·D18·D19·D21·D22·D23·D24·D25·D26·D27·D28·D29·D31·D32·D33·D35·D36·D37·D39
C10	Z00·Z05·Z06·Z08·Z09·Z10·Z11·Z14·Z15· D00·D01·D04·D05·D06·D07·D09·D10·D15·D16·D17·D18·D19·D20·D22·D23·D24·D25·D26·D27·D28·D29·D30·D32·D33·D34·D37·D36·D38·D40
C09	Z04·Z05·Z07·Z08·Z09·Z10·Z13·Z14·Z15· D00·D01·D02·D05·D06·D07·D08·D10·D11·D16·D17·D18·D19·D20·D21·D23·D24·D25·D26·D27·D28·D29·D30·D31·D33·D34·D35·D37·D38·D39·D41
C08	Z03·Z04·Z06·Z07·Z08·Z09·Z12·Z13·Z14· D01·D02·D03·D06·D07·D08·D09·D11·D12·D17·D18·D19·D20·D21·D22·D24·D25·D26·D27·D28·D29·D30·D31·D32·D34·D35·D36·D38·D39·D40·D42
C07	Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13· D02·D03·D04·D07·D08·D09·D10·D12·D13·D18·D19·D20·D21·D22·D23·D25·D26·D27·D28·D29·D30·D31·D32·D33·D35·D36·D37·D39·D40·D41·D43
C06	Z01·Z02·Z04·Z05·Z06·Z07·Z10·Z11·Z12· D03·D04·D05·D08·D09·D10·D11·D13·D14·D19·D20·D21·D22·D23·D24·D26·D27·D28·D29·D30·D31·D32·D33·D34·D36·D37·D38·D40·D41·D42·D44
C05	Z00·Z01·Z03·Z04·Z05·Z06·Z09·Z10·Z11· D04·D05·D06·D09·D10·D11·D12·D14·D15·D20·D21·D22·D23·D24·D25·D27·D28·D29·D30·D31·D32·D33·D34·D35·D37·D38·D39·D41·D42·D43·D45
C04	Z00·Z02·Z03·Z04·Z05·Z08·Z09·Z10·Z15·D00·D05·D06·D07·D10·D11·D12·D13·D15·D16·D21·D22·D23·D24·D25·D26·D28·D29·D30·D31·D32·D33·D34·D35·D36·D38·D39·D40·D42·D43·D44·D46
C03	Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14· D01·D06·D07·D08·D11·D12·D13·D14·D16·D17·D22·D23·D24·D25·D26·D27·D29·D30·D31·D32·D33·D34·D35·D36·D37·D39·D40·D41·D43·D44·D45·D47
C02	Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12· D03·D04·D05·D08·D09·D11·D12·D13·D15·D16·D18·D19·D20·D22·D23·D24·D26·D30·D31·D32·D34·D37·D38·D41·D42·D45·D46·
C01	Z01·Z02·Z03·Z05·Z06·Z09·Z10·Z11·Z15· D00·D04·D05·D06·D09·D10·D12·D13·D14·D16·D17·D19·D20·D21·D23·D24·D25·D27·D31·D32·D33·D35·D38·D39·D42·D43·D46·D47
C00	Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14· D01·D02·D03·D04·D06·D10·D13·D15·D16·D18·D19·D21·D24·D26·D27·D32·D34·D35·D39·D43·D47

"Z" denotes an initial value of latch (flip-flop) L.

c. Replacement of data Page 8

Operational expressions described earlier are substituted into operational expressions obtained in "b", since D47-D31 are the CRC16 ① arithmetic operation results, as clear from "a".

Replacement example in the least significant bit (CO)

C0= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
D01·D02·D03·D04·D06·D10·D13·D15·D16·D18·D19·D21·D24·D26·D27·D32·D34·
D35·D39·D43·D47

P7

D47 <= C15= X03·X04·X06·X09·X11·X12·X14·X15·
D00·D01·D03·D04·D06·D09·D11·D12·D17·D19·D20·D24·D28

D43 <= C11= X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15·
D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·
D17·D19·D20·D21·D23

D39 <= C07= X00·X01·X02·X03·X04·X05·X06·X08·X09·X10·X11·X12·X13·
D02·D03·D04·D05·D06·D07·D09·D10·D11·D12·D13·D14·D15·D16·D17·
D19·D20·D21·D23·D24·D25·D27

D35 <= C03= X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15·
D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·
D20·D21·D23·D24·D25·D27·D28·D29·D31

D34 <= C02= X00·X01·X05·X07·X08·X09·X11·X12·X13·X15·
D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·
D26·D29·D30

D32 <= C00= X04·X05·X07·X10·X12·X13·X15·
D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

Substituting

C0= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
D01·D02·D03·D04·D06·D10·D13·D15·D16·D18·D19·D21·D24·D26·D27·
X03·X04·X06·X09·X11·X12·X14·X15·
D00·D01·D03·D04·D06·D09·D11·D12·D17·D19·D20·D24·D28·
X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15·
D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·
D19·D20·D21·D23·
X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15·
D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·
D20·D21·D23·D24·D25·D27·D28·D29·D31·
X00·X01·X05·X07·X08·X09·X11·X12·X13·X15·
D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·
D26·D29·D30·
X04·X05·X07·X10·X12·X13·X15·
D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

Deleting same terms

C0= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
X01·X02·X03·X04·X05·X09·X10·X11·X12·X14·X15·
D00·D02·D05·D11·D12·D14·D15·D21·D22·D25·D30

d. Operational expressions

It is possible to obtain CRC16 arithmetic operation device ②' by performing the above processing on all the operational expressions.

	Operational expressions
C15	Z01·Z04·Z08·Z10·Z11·Z12·Z13· X00·X02·X03·X04·X05·X06·X10·X11·X12·X13·X15· D00·D07·D09·D10·D12·D13·D14·D15·D16·D18·D19·D20·D21·D22·D23·D24·D26·D27·D28· D29·D31
C14	Z00·Z03·Z07·Z09·Z10·Z11·Z12· X01·X03·X04·X05·X06·X07·X11·X12·X13·X14· D01·D02·D05·D06·D09·D10·D11·D14·D15·D16·D18·D21·D22·D24·D25·D26·D29· D30
C13	Z02·Z06·Z08·Z09·Z10·Z11·Z15· X00·X02·X06·X08·X10·X14· D00·D01·D04·D06·D15·D16·D19·D27·D30·D31
C12	Z01·Z05·Z07·Z08·Z09·Z10·Z14· X01·X03·X07·X09·X11·X15· D00·D01·D04·D05·D07·D10·D12·D19·D21·D22·D25·D31
C11	Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15· X00·X03·X04·X06·X07·X08·X10·X11·X12· D00·D06·D07·D11·D12·D14·D16·D18·D19·D20·D23·D27·D28·D29
C10	Z00·Z05·Z06·Z08·Z09·Z10·Z11·Z14·Z15· X01·X04·X05·X07·X08·X09·X11·X12·X13· D00·D01·D02·D03·D05·D08·D09·D11·D14·D15·D17·D18·D21·D22·D26·D27·D28·D29
C09	Z04·Z05·Z07·Z08·Z09·Z10·Z13·Z14·Z15· X00·X02·X05·X06·X08·X09·X10·X12·X13·X14· D00·D03·D08·D09·D11·D13·D15·D16·D17·D21·D22·D25·D26·D27·D28·D30·D31
C08	Z03·Z04·Z06·Z07·Z08·Z09·Z12·Z13·Z14· X01·X03·X06·X07·X09·X10·X11·X13·X14·X15· D00·D03·D04·D05·D07·D11·D14·D20·D23·D24·D25·D26·D27·D29·D30·D31
C07	Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13· X02·X05·X08·X11·X13·X14· D01·D03·D08·D09·D12·D17·D18·D25·D26·D27·D28·D29·D30
C06	Z01·Z02·Z04·Z05·Z06·Z07·Z10·Z11·Z12· X03·X04·X05·X06·X07·X09·X10·X13·X14· D01·D02·D03·D04·D06·D12·D13·D14·D19·D24·D26·D28·D29
C05	Z00·Z01·Z03·Z04·Z05·Z06·Z09·Z10·Z11· X06·X08·X11·X12·X13·X14· D01·D02·D03·D05·D06·D07·D10·D11·D12·D14·D15·D16·D18·D24·D25·D26·D28
C04	Z00·Z02·Z03·Z04·Z05·Z08·Z09·Z10·Z15· X00·X07·X09·X12·X13·X14·X15· D01·D02·D03·D05·D07·D08·D10·D11·D12·D13·D16·D17·D19·D20·D23·D24·D31
C03	Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14· X01·X08·X10·X13·X14·X15· D00·D02·D05·D06·D08·D11·D12·D13·D17·D18·D19·D20·D21·D22·D23·D26·D28·D30·D31
C02	Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12· X03·X06·X07·X09·X14·X15· D00·D01·D03·D04·D05·D06·D11·D13·D15·D19·D21·D22·D24·D25·D26·D28·D29·D30
C01	Z01·Z02·Z03·Z05·Z06·Z09·Z10·Z11·Z15· X00·X05·X08·X12·X13· D00·D02·D03·D04·D05·D06·D07·D09·D12·D13·D14·D15·D18·D19·D21·D22·D25·D27·D28· D30
C00	Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14· X01·X02·X03·X04·X05·X09·X10·X11·X12·X14·X15· D00·D02·D05·D11·D12·D14·D15·D21·D22·D25·D30

"X" denotes an initial value of latch (flip-flop) H.
"Z" denotes an initial value of latch (flip-flop) L.

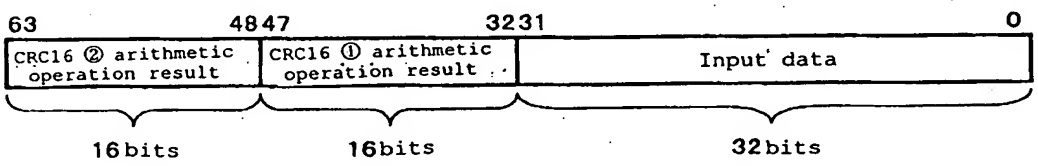
③ CRC16 arithmetic operation device ③′ U:
Operational Expression Production

a. Data being of 64 bits in length.

In the CRC16 arithmetic operation device ③′ S, the operational expression is produced using input data (32 bits) and an immediately preceding arithmetic operation result (16 bits).

With the conventional method, CRC16 code bit is acquired by adding the CRC16 code bit (result obtained from CRC16 arithmetic operation device ②: 16 bits) to an end part of the input data.

The CRC16 arithmetic operation device ③′ S incorporated in the present invention inputs simultaneously the input data, CRC16 ① arithmetic operation result (obtained two operations before a final CRC16 arithmetic operation result), and CRC16 ②′ arithmetic operation result (obtained one operation before a final CRC16 arithmetic operation result). That is, the operational expression is produced as 64 bits of the input data. At this stage, original input data (D) as lower-order bits and the CRC arithmetic operation result as higher-order bits (see below) are respectively input.



b. Operational Expression Production-I

First, a CRC16 operational expression on input data being of 64 bits in length is produced. This is the output data from each of the flip-flops making up the CRC16 generating circuit described in the previous document, when 64 bits of data is shifted. on page 6

	Operational expressions
C15	R01·R03·R04·R07·R08·R10·R11·R12·R13·R15·D00·D02·D03·D04·D05·D07·D08·D11·D12·D14·D18·D19·D20·D21·D23·D27·D30·D32·D33·D35·D36·D38·D41·D43·D44·D49·D51·D52·D56·D60
C14	R00·R02·R03·R06·R07·R09·R10·R11·R12·R14·R15·D00·D01·D03·D04·D05·D06·D08·D09·D12·D13·D15·D19·D20·D21·D22·D24·D28·D31·D33·D34·D36·D37·D39·D42·D44·D45·D50·D52·D53·D57·D61
C13	R01·R02·R05·R06·R08·R09·R10·R11·R13·R14·R15·D00·D01·D02·D04·D05·D06·D07·D09·D10·D13·D14·D16·D20·D21·D22·D23·D25·D29·D32·D34·D35·D37·D38·D40·D43·D45·D46·D51·D53·D54·D58·D62
C12	R00·R01·R04·R05·R07·R08·R09·R10·R12·R13·R14·R15·D00·D01·D02·D03·D05·D06·D07·D08·D10·D11·D14·D15·D17·D21·D22·D23·D24·D26·D30·D33·D35·D36·D38·D39·D41·D44·D46·D47·D52·D54·D55·D59·D63
C11	R00·R01·R06·R09·R10·R14·D01·D05·D06·D09·D14·D15·D16·D19·D20·D21·D22·D24·D30·D31·D32·D33·D34·D35·D37·D38·D39·D40·D41·D42·D43·D44·D45·D47·D48·D49·D51·D52·D53·D55
C10	R00·R05·R08·R09·R13·D02·D06·D07·D10·D15·D16·D17·D20·D21·D22·D23·D25·D26·D31·D32·D33·D34·D35·D36·D38·D39·D40·D41·D42·D43·D44·D45·D46·D48·D49·D50·D52·D53·D54·D56
C09	R04·R07·R08·R12·R15·D00·D03·D07·D08·D11·D16·D17·D18·D21·D22·D23·D24·D26·D27·D32·D33·D34·D35·D36·D37·D39·D40·D41·D42·D43·D44·D45·D46·D47·D49·D50·D51·D53·D54·D55·D57
C08	R03·R06·R07·R11·R14·R15·D00·D01·D04·D08·D09·D12·D17·D18·D19·D22·D23·D24·D25·D27·D28·D33·D34·D35·D36·D37·D38·D40·D41·D42·D43·D44·D45·D46·D47·D48·D50·D51·D52·D54·D55·D56·D58
C07	R02·R05·R06·R10·R13·R14·R15·D00·D01·D02·D05·D09·D10·D13·D18·D19·D20·D23·D24·D25·D26·D28·D29·D34·D35·D36·D37·D38·D39·D41·D42·D43·D44·D45·D46·D47·D48·D49·D51·D52·D53·D55·D56·D57·D59
C06	R01·R04·R05·R09·R12·R13·R14·D01·D02·D03·D06·D10·D11·D14·D19·D20·D21·D24·D25·D26·D27·D29·D30·D35·D36·D37·D38·D39·D40·D42·D43·D44·D45·D46·D47·D48·D49·D50·D52·D53·D54·D56·D57·D58·D60
C05	R00·R03·R04·R08·R11·R12·R13·D02·D03·D04·D07·D11·D12·D15·D20·D21·D22·D25·D26·D27·D28·D30·D31·D36·D37·D38·D39·D40·D41·D43·D44·D45·D46·D47·D48·D49·D50·D51·D53·D54·D55·D57·D58·D59·D61
C04	R02·R03·R07·R10·R11·R12·R15·D00·D03·D04·D05·D08·D12·D13·D16·D21·D22·D23·D26·D27·D28·D29·D31·D32·D37·D38·D39·D40·D41·D42·D44·D45·D46·D47·D48·D49·D50·D51·D52·D54·D55·D56·D58·D59·D60·D62
C03	R01·R02·R06·R09·R10·R11·R14·D01·D04·D05·D06·D09·D13·D14·D17·D22·D23·D24·D27·D28·D29·D30·D32·D33·D38·D39·D40·D41·D42·D43·D45·D46·D47·D48·D49·D50·D51·D52·D53·D55·D56·D57·D59·D60·D61·D63
C02	R00·R03·R04·R05·R07·R09·R11·R12·D03·D04·D06·D08·D10·D11·D12·D15·D19·D20·D21·D24·D25·D27·D28·D29·D31·D32·D34·D35·D36·D38·D39·D40·D42·D46·D47·D48·D50·D53·D54·D57·D58·D61·D62
C01	R02·R03·R04·R06·R08·R10·R11·R15·D00·D04·D05·D07·D09·D11·D12·D13·D16·D20·D21·D22·D25·D26·D28·D29·D30·D32·D33·D35·D36·D37·D39·D40·D41·D43·D47·D48·D49·D51·D54·D55·D58·D59·D62·D63
C00	R02·R04·R05·R08·R09·R11·R12·R13·R14·D01·D02·D03·D04·D06·D07·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63

"R" denotes an initial value of latch (flip-flop) R.

c. Replacement of data Page 12 page 8

Operational expressions described earlier are substituted into operational expressions obtained in "b", since D63-D48 are the CRC16 ②' arithmetic operation results, as clear from "a".

Operational expressions described earlier are substituted into operational expressions obtained in "b", since D47-D32 are the CRC16 ① arithmetic operation results, as clear from "a".

Replacement example in the least significant bit (CO)

C0= R02·R04·R05·R08·R09·R11·R12·R13·R14·
D01·D02·D03·D04·D06·D07·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63

P11

D63 <= C15= Z01·Z04·Z08·Z10·Z11·Z12·Z13·
X00·X02·X03·X04·X05·X06·X10·X11·X12·X13·X15·
D00·D07·D09·D10·D12·D13·D14·D15·D16·D18·D19·D20·D21·D22·D23·D24·
D26·D27·D28·D29·D31

D59 <= C11= Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15·
X00·X03·X04·X06·X07·X08·X10·X11·X12·
D00·D06·D07·D11·D12·D14·D16·D18·D19·D20·D23·D27·D28·D29

D55 <= C07= Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13·X02·X05·X08·X11·X13·
X14·
D01·D03·D08·D09·D12·D17·D18·D25·D26·D27·D28·D29·D30

D51 <= C03= Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14·X01·X08·X10·X13·X14·X15·
D00·D02·D05·D06·D08·D11·D12·D13·D17·D18·D19·D20·D21·D22·D23·D26·
D28·D30·D31

D50 <= C02= Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12·X03·X06·X07·X09·X14·
X15·
D00·D01·D03·D04·D05·D06·D11·D13·D15·D19·D21·D22·D24·D25·D26·D28·
D29·D30

D48 <= C00= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
X01·X02·X03·X04·X05·X09·X10·X11·X12·X14·X15·
D00·D02·D05·D11·D12·D14·D15·D21·D22·D25·D30

P7

D43 <= C11= X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15·
D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·
D19·D20·D21·D23

D42 <= C10= X01·X02·X03·X04·X05·X06·X07·X08·X09·X11·X12·X13·X14·X15·
D00·D01·D02·D03·D04·D06·D07·D08·D09·D10·D11·D12·D13·D14·D16·D17·
D18·D20·D21·D22·D24

D40 <= C08= X00·X01·X02·X03·X04·X05·X06·X07·X09·X10·X11·X12·X13·X14·
D01·D02·D03·D04·D05·D06·D08·D09·D10·D11·D12·D13·D14·D15·D16·D18·
D19·D20·D22·D23·D24·D26

D37 <= C05= X00·X01·X02·X03·X04·X06·X07·X08·X09·X10·X11·
D04·D05·D06·D07·D08·D09·D11·D12·D13·D14·D15·D16·D17·D18·D19·D21·
D22·D23·D25·D26·D27·D29

D35 <= C03= X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15·
D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·D20·
D21·D23·D24·D25·D27·D28·D29·D31

D34 <= C02= X00·X01·X05·X07·X08·X09·X11·X12·X13·X15·
D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·D26·
D29·D30

D32 <= C00= X04·X05·X07·X10·X12·X13·X15
D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

Substituting

C0= Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·

C0= R02·R04·R05·R08·R09·R11·R12·R13·R14·
D01·D02·D03·D04·D06·D07·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
Z01·Z04·Z08·Z10·Z11·Z12·Z13·
X00·X02·X03·X04·X05·X06·X10·X11·X12·X13·X15·
D00·D07·D09·D10·D12·D13·D14·D15·D16·D18·D19·D20·D21·D22·D23·D24·
D26·D27·D28·D29·D31·
Z00·Z01·Z06·Z07·Z09·Z10·Z11·Z12·Z15·
X00·X03·X04·X06·X07·X08·X10·X11·X12·
D00·D06·D07·D11·D12·D14·D16·D18·D19·D20·D23·D27·D28·D29·
Z02·Z03·Z05·Z06·Z07·Z08·Z11·Z12·Z13·X02·X05·X08·X11·X13·X14·
D01·D03·D08·D09·D12·D17·D18·D25·D26·D27·D28·D29·D30·
Z01·Z02·Z03·Z04·Z07·Z08·Z09·Z14·X01·X08·X10·X13·X14·X15·
D00·D02·D05·D06·D08·D11·D12·D13·D17·D18·D19·D20·D21·D22·D23·D26·
D28·D30·D31·
Z00·Z02·Z03·Z04·Z06·Z07·Z10·Z11·Z12·X03·X06·X07·X09·X14·X15·
D00·D01·D03·D04·D05·D06·D11·D13·D15·D19·D21·D22·D24·D25·D26·D28·
D29·D30·
Z00·Z02·Z05·Z09·Z11·Z12·Z13·Z14·
X01·X02·X03·X04·X05·X09·X10·X11·X12·X14·X15·
D00·D02·D05·D11·D12·D14·D15·D21·D22·D25·D30·
X00·X02·X03·X04·X05·X06·X07·X08·X09·X10·X12·X13·X14·X15·
D00·D01·D02·D03·D05·D06·D07·D08·D09·D10·D11·D12·D13·D15·D16·D17·
D19·D20·D21·D23·
X01·X02·X03·X04·X05·X06·X07·X08·X09·X11·X12·X13·X14·X15·
D00·D01·D02·D03·D04·D06·D07·D08·D09·D10·D11·D12·D13·D14·D16·D17·
D18·D20·D21·D22·D24·
X00·X01·X02·X03·X04·X05·X06·X07·X09·X10·X11·X12·X13·X14·
D01·D02·D03·D04·D05·D06·D08·D09·D10·D11·D12·D13·D14·D15·D16·D18·
D19·D20·D22·D23·D24·D26·
X00·X01·X02·X03·X04·X06·X07·X08·X09·X10·X11·
D04·D05·D06·D07·D08·D09·D11·D12·D13·D14·D15·D16·D17·D18·D19·D21·
D22·D23·D25·D26·D27·D29·
X00·X01·X02·X04·X05·X06·X07·X08·X09·X14·X15·
D00·D01·D06·D07·D08·D09·D10·D11·D13·D14·D15·D16·D17·D18·D19·D20·
D21·D23·D24·D25·D27·D28·D29·D31·
X00·X01·X05·X07·X08·X09·X11·X12·X13·X15·
D00·D02·D03·D04·D06·D07·D08·D10·D14·D15·D16·D18·D21·D22·D25·D26·
D29·D30·
X04·X05·X07·X10·X12·X13·X15
D00·D02·D03·D05·D08·D10·D11·D16·D18·D19·D23·D27·D31

Deleting same terms

C0= R02·R04·R05·R08·R09·R11·R12·R13·R14·
Z02·Z03·Z04·Z07·Z09·Z10·Z11·
X02·X04·X05·X06·X08·X12·X13·
D01·D05·D08·D09·D11·D12·D13·D16·D17·D18·D21·D22·D24·D30·D31

d. Operational expressions

It is possible to obtain CRC16 arithmetic operation device ③′ by performing the above processing on all the operational expressions.

	Operational expressions
C15	R01·R03·R04·R07·R08·R10·R11·R12·R13·R15·Z03·Z04·Z05·Z08·Z10·Z11·Z12·X01·X05·X06·X07·X09·X14·D00·D01·D02·D06·D09·D16·D19·D20·D23·D24·D26·D28
C14	R00·R02·R03·R06·R07·R09·R10·R11·R12·R14·R15·Z00·Z04·Z05·Z06·Z09·Z11·Z12·Z13·X00·X03·X05·X08·X09·X11·X12·X13·X14·D00·D03·D04·D05·D06·D07·D08·D11·D13·D15·D17·D18·D19·D24·D27·D28·D29·D30
C13	R01·R02·R05·R06·R08·R09·R10·R11·R13·R14·R15·Z00·Z01·Z02·Z06·Z07·Z09·Z10·Z11·X01·X04·X05·X06·X07·X08·X11·X12·X13·X15·D01·D03·D04·D09·D11·D14·D15·D18·D20·D21·D22·D23·D31
C12	R00·R01·R04·R05·R07·R08·R09·R10·R12·R13·R14·R15·Z00·Z01·Z02·Z03·Z04·Z05·Z08·Z11·Z13·Z15·X00·X03·X04·X05·X06·X07·X10·X11·X12·X14·D06·D09·D10·D11·D13·D15·D21·D22·D24·D26·D31
C11	R00·R01·R06·R09·R10·R14·Z00·Z01·Z04·Z06·Z07·Z08·Z15·X03·X04·X05·X06·X07·X11·X13·X14·X15·D02·D04·D05·D06·D07·D09·D10·D12·D16·D20·D21·D22·D23·D24·D25·D26·D28·D31
C10	R00·R05·R08·R09·R13·Z01·Z02·Z04·Z08·Z09·Z10·Z12·Z13·Z15·X02·X05·X09·X10·X11·X13·D00·D02·D03·D04·D06·D11·D14·D15·D16·D18·D19·D21·D22·D24·D26·D28·D31
C09	R04·R07·R08·R12·R15·Z02·Z03·Z05·Z09·Z10·Z11·Z13·Z14·X01·X04·X08·X09·X10·X12·D00·D01·D02·D04·D08·D10·D12·D13·D14·D16·D17·D19·D25·D26·D27·D29·D30
C08	R03·R06·R07·R11·R14·R15·Z03·Z04·Z06·Z10·Z11·Z12·Z14·Z15·X00·X04·X06·X07·X08·X12·X14·X15·D00·D01·D05·D07·D09·D15·D18·D21·D22·D24·D25·D26·D27·D29·D30
C07	R02·R05·R06·R10·R13·R14·R15·Z00·Z02·Z05·Z09·Z10·Z12·Z13·Z14·X01·X02·X03·X07·X10·X11·X13·X14·D00·D01·D02·D03·D04·D05·D06·D08·D09·D12·D13·D14·D15·D17·D19·D26·D28·D30
C06	R01·R04·R05·R09·R12·R13·R14·Z00·Z01·Z02·Z03·Z04·Z06·Z07·Z09·Z13·X00·X03·X04·X05·X06·X11·X13·X14·X15·D00·D02·D03·D04·D08·D11·D13·D16·D17·D20·D21·D22·D23·D24·D25
C05	R00·R03·R04·R08·R11·R12·R13·Z00·Z01·Z03·Z04·Z07·Z08·Z09·Z10·Z11·Z12·Z13·X01·X09·X11·X13·X15·D00·D02·D04·D05·D08·D17·D20·D23·D26·D27·D28·D29·D31
C04	R02·R03·R07·R10·R11·R12·R15·Z01·Z02·Z07·Z08·Z09·Z11·Z14·Z15·X00·X03·X04·X06·X08·X09·X10·X11·D01·D03·D04·D09·D11·D14·D15·D18·D20·D21·D22·D23·D31
C03	R01·R02·R06·R09·R10·R11·R14·Z00·Z02·Z03·Z04·Z05·Z07·Z08·Z09·Z13·X02·X04·X05·X06·X07·X08·X10·X11·X12·X14·X15·D00·D02·D03·D12·D13·D14·D15·D16·D24·D26·D27·D28·D29·D30·D31
C02	R00·R03·R04·R05·R07·R09·R11·R12·Z00·Z01·Z02·Z04·Z06·Z07·Z10·Z12·Z15·X01·X02·X05·X09·X12·X13·X14·D00·D01·D02·D03·D04·D05·D09·D12·D18·D20·D23·D25·D30·D31
C01	R02·R03·R04·R06·R08·R10·R11·R15·Z01·Z03·Z07·Z08·Z09·Z12·Z14·X00·X02·X03·X05·X08·X09·X10·X13·X14·X15·D02·D03·D04·D07·D08·D09·D10·D11·D12·D14·D15·D17·D18·D19·D25·D26·D27·D28·D29·D31·
C00	R02·R04·R05·R08·R09·R11·R12·R13·R14·Z02·Z03·Z04·Z07·Z09·Z10·Z11·X02·X04·X05·X06·X08·X12·X13·D01·D05·D08·D09·D11·D12·D13·D16·D17·D18·D21·D22·D24·D30·D31

"R" denotes an initial value of latch (flip-flop) R.
"X" denotes an initial value of latch (flip-flop) H.
"Z" denotes an initial value of latch (flip-flop) L.

Replacement example in the least significant bit (CO)

C0= R2·R4·R5·R8·R9·R11·R12·R13·R14·
D1·D2·D3·D4·D6·D7·D10·D11·D13·D17·D18·D19·D20·D22·D26·D29·D31·
D32·D34·D35·D37·D40·D42·D43·D48·D50·D51·D55·D59·D63

P7, 8

D63 <= C31= R5·R8·R9·R11·R15·R23·R24·R25·R27·R28·R29·R30·R31·
D0·D1·D2·D3·D4·D6·D7·D8·D16·D20·D22·D23·D26

D59 <= C27= R1·R4·R5·R7·R11·R19·R20·R21·R23·R24·R25·R26·R27·
R29·
D2·D4·D5·D6·D7·D8·D10·D11·D12·D20·D24·D26·D27·D30

D55 <= C23= R0·R1·R6·R9·R13·R15·R16·R17·R19·R20·R26·R27·R29·
R31·
D0·D2·D4·D5·D11·D12·D14·D15·D16·D18·D22·D25·D30·D31

D51 <= C19= R3·R7·R8·R11·R15·R16·R20·R22·R24·R25·R27·R29·
D2·D4·D6·D7·D9·D11·D15·D16·D20·D23·D24·D28

D50 <= C18= R2·R6·R7·R10·R14·R15·R19·R21·R23·R24·R26·R28·R31·
D0·D3·D5·D7·D8·D10·D12·D16·D17·D21·D24·D25·D29

D48 <= C16= R0·R4·R5·R8·R12·R13·R17·R19·R21·R22·R24·R26·R29·
R30·
D1·D2·D5·D7·D9·D10·D12·D14·D18·D19·D23·D26·D27·D31

D43 <= C11= R0·R1·R3·R4·R9·R12·R14·R15·R16·R17·R20·R24·R25·
R26·R27·R28·R31·
D0·D3·D4·D5·D6·D7·D11·D14·D15·D16·D17·D19·D22·D27·D28·D30·
D31

D42 <= C10= R0·R2·R3·R5·R9·R13·R14·R16·R19·R26·R28·R29·R31·
D0·D2·D3·D5·D12·D15·D17·D18·D22·D26·D28·D29·D31

D40 <= C8= R0·R1·R3·R4·R8·R10·R11·R17·R22·R28·R31·
D0·D3·D8·D9·D14·D19·D20·D21·D23·D27·D28·D30·D31

D37 <= C5= R0·R1·R3·R4·R5·R6·R7·R10·R13·R19·R20·R21·R24·R28·R29·
D2·D3·D7·D10·D11·D12·D18·D21·D24·D25·D26·D27·D28·D30·D31

D35 <= C3= R1·R2·R3·R7·R8·R9·R10·R14·R15·R17·R18·R19·R25·R27·R31·
D0·D4·D6·D12·D13·D14·D16·D17·D21·D22·D23·D24·D28·D29·D30

D34 <= C2= R0·R2·R6·R7·R8·R9·R13·R14·R16·R17·R18·R24·R26·R30·R31·
D0·D1·D5·D7·D13·D14·D15·D17·D18·D22·D23·D24·D25·D29·D30·
D31

D32 <= C0= R0·R6·R9·R10·R12·R16·R24·R25·R26·R28·R29·R30·R31·
D0·D1·D2·D3·D5·D6·D7·D15·D19·D21·D22·D25·D31